Summer 7-16-2014

Predicting the Performance of Rosetta Stone Language Learners with Individualized Models of Forgetting

Karl Ridgeway
University of Colorado Boulder, karl.ridgeway@gmail.com

Predicting the Performance of Rosetta Stone® Language Learners with Individualized Models of Forgetting

by

Karl Ridgeway

B.S., James Madison University, 2005

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Computer Science

Department of Computer Science

2014

This thesis entitled:
Predicting the Performance of Rosetta Stone® Language Learners with Individualized Models of Forgetting
written by Karl Ridgeway
has been approved for the Department of Computer Science

_____
Prof. Mike Mozer

_____
Prof. Matt Jones

_____
Prof. Clayton Lewis

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

, Karl Ridgeway ( )

Predicting the Performance of Rosetta Stone® Language Learners with Individualized Models of Forgetting

Thesis directed by Prof. Mike Mozer

I explore the nature of forgetting in a corpus of 125,000 students using the Rosetta Stone® foreign-language instruction software on 48 Spanish lessons. Students are tested on a lesson after its completion and are then retested after a variable time lag. The observed power-law forgetting curves have a small temporal decay rate that varies from lesson to lesson. I obtain improved predictive accuracy of the forgetting model by augmenting it with features that encode characteristics of a student's initial study of the lesson and the activities the student engaged in between the two tests. I then analyze which features best explain individual performance, and find that using these features the augmented model can predict about 25% of the variance in an individual's score on the second test.

Contents

Chapter

List of Tables

Table

List of Figures

Figure

0.1 Introduction

0.1.1 Forgetting

Psychologists have studied the durability of memory over time, or forgetting, for almost 130 years. Hermann Ebbinghaus first formalized forgetting as an exponential forgetting curve, in his 1885 work "Über das Gedächtnis" ("On Memory") [8], using himself as the only experimental subject. Ebbinghaus measured his own ability to recall associations of meaning with "nonsense syllables" over varying lengths of time.

Since then, lab researchers have updated and refined this model, introducing other variations, such as the power-law forgetting curve [17]. Laboratory memory experiments, such as those performed by Wickelgren [15], typically involve few subjects and short retention intervals, given that the tested material is also learned in the laboratory setting. Wickelgren's experiments, for example, typically tested subjects on material over retention intervals from several seconds to several weeks. These experiments also typically involved mostly undergraduate students.

Forgetting has been studied not only in the lab but also in more naturalistic settings. Evidence from medical education research [6] shows that medical students forget 25-35% of basic science knowledge after a single year, 50% after two years, and up to 85% after 25 years [7].

In addition, Harry P. Bahrick and colleagues have conducted many studies evaluating memory strength over a long time scale [15]. The Bahrick experiments tested on a wide range of real-world knowledge of participants over intervals from 3 months to 50 years. The knowledge came from a wide range of life experience such as material studied in school years ago to names of classmates. The Bahrick data, therefore, are made up of a wide range of subject ages, as one must be at least 50 years old to recall something learned 50 years ago. Another interesting attribute of the Bahrick studies is that they often tested individuals on material that had likely been restudied in an uncontrolled way throughout the subjects' lifetimes.

In [1], Bahrick studied the ability of 851 current and former students of Ohio Wesleyan University to recall spacial information about Delaware, OH, where the college is located. The

subjects ages ranged from college-age through late adulthood, and the retention intervals ranged from 0-46 years, 4 months. The tests consisted of a number of tasks: free recall of street names, free recall of buildings and landmarks, visually cued recall, verbally cued recall, and matching. This study showed that forgetting of campus landmarks, regardless of testing method, dropped rapidly in the first several years after graduation and then continued to slowly decay over longer retention intervals.

Studies with long retention intervals can show a strong effect of forgetting over time, even with good initial performance on the material. A study by Bahrick and Hall [3], looked at a group of 1726 students who studied high school mathematics. The retention intervals ranged from 0 to 74 years. The mathematical subjects studied were algebra and plane geometry. Both are subjects that did not undergo significant changes during the time span of the tested retention intervals. The study examined four groups of students who took an increasing number of mathematics classes (although the test only assessed proficiency at algebra and plane geometry) - those who took only one algebra class, those who took more than one algebra class, those took took calculus, and those who took even more advanced classes. The study showed that students who went on to take calculus classes forgot the algebra material at a lower rate than those who stopped after algebra.

The effect of forgetting applies across all types of knowledge, including second language knowledge. Notably, Ebbinghaus's self-experiments examined the retention of linguistic knowledge, albeit for nonsense syllables. Studies of forgetting with respect to second language acquisition typically focus on recall of vocabulary words, for which assessments are easy to construct.

Forgetting foreign language vocabulary happens quickly, but can be offset by gradually learning material over a period of time instead of in one long session. A study of 56 high school students learning french vocabulary words showed strong evidence of forgetting after just a few days [5]. The students were placed into two groups - one practiced a set of french words for 30 minutes on one day, the other practiced the words in three 10-minute sessions over the course of three successive days. The students were tested on the same material 4 days later. The group that practiced in distributed manner, in three 10-minute sessions, performed better at the end of the retention in-

terval than the massed practice group. This practice strategy takes advantage of the spacing effect - a well-known effect where distributed practice leads to better retention over time. However, both groups showed evidence of forgetting after only a few days.

The spacing effect also holds true for longer inter-session intervals and longer retention intervals. In an almost decade-long longitudinal investigation, Bahrick showed a strong effect of forgetting of second-language vocabulary words over many years [2]. In that study, a small group of 4 subjects learned foreign language words (french and german) and were tested on the material after 1, 2, 3, and 5 years. The goal of the study was to find evidence of the spacing effect, which refers to a lower rate of forgetting for material practiced with inter-session intervals. Two training policies were developed, which varied by their inter-session interval: 14 days, 28 days, and 56 days. The amount of training was held constant across policies, so each policy took successively longer to execute. Each subject studied some words with each policy. The longer inter-session intervals had a noticeable positive effect on retention over time.

The forgetting effect in foreign-language knowledge is not limited to vocabulary retention. Other types of language learning, such as perceptual phonetic training, can also show evidence of weakening effects over time. In one study, 19 native Japanese speakers were trained to discriminate the English /ɹ/ vs /l/ phonetic contrast [13]. The subjects showed an improvement in discrimination of the contrast which held after 3 months, but showed decay in contrast discrimination after 6 months.

In [15], Rubin lays out a framework for models of forgetting and the data used to fit them tend to vary in several ways:

(1) Type of function used to describe recall over time (e.g. linear, hyperbolic, exponential, power)

(2) Whether the data are aggregated

   (a) Aggregated by student

   (b) Aggregated by item

(3) At what level the forgetting function is specified

   (a) Individualized to the student

   (b) Specific to the content

   (c) Specific to some other cross-cutting property of the student, such as age

   (d) Specific to some test procedure. For example, the test could be simple recognition, multiple choice, or free response

   (e) Specific to a particular scale of retention time interval

In this work, I use power-law forgetting functions and alternative models to predict student performance in a large language learning data set from the Rosetta Stone® course. These forgetting models will be individualized to the student, and specific to the content being studied. They will also be able to characterize forgetting at time intervals ranging from minutes to years. The content being studied consists of many types of language learning knowledge, for example: vocabulary, syntactic knowledge, morphological knowledge such as inflections and derivations, or even phonetic perception and production knowledge. Furthermore, the models built in this work will be agnostic of cross-cutting features of learners such as age, gender, or location.

0.1.1.1 Three- and Two-Parameter Power-Law Forgetting Models

The exponential function that Ebbinghaus developed is characterized by a a constantly decelerating rate of decay of memory over time. The current consensus is that memory strength of some material over time can be described by a power law function [17], which is characterized by an initial rapid drop in performance, followed by a long period of very slow decay. While there are many other functions that can produce good fits [15], the power law function fits well to varying retention interval scales, has easily interpretable coefficients, and can fit both aggregated and individual data. These power-law forgetting functions are all characterized by a sharp drop in recall over a short amount of time, followed by a gradual decay over longer time scales. A common formulation [17] of this power law is

$$Pr(\text{recall}) = \alpha(1 + \gamma t)^{\beta} \tag{1}$$

where α is the state of knowledge at $t = 0$, γ is a scaling factor on time, and β is the rate of forgetting (a negative exponent). Equation 1 has the "1+" term in order to handle the case of $t = 0$. As $gamma * t$ becomes large, the "1+" term becomes unimportant, and the γ term becomes redundant with α. Consequently, the model can be reduced to a two parameter model (based on the formulation in [17]):

$$Pr(\text{recall}) = \alpha t^{\beta} \tag{2}$$

In this formulation, α and the intercept term are removed. It can be considered an approximation to 1, but has the drawback of not estimating the degree of initial learning directly. According to [17], these functions describe forgetting over entire populations of individuals, populations of items, or both.

0.1.2 Forgetting In A Massively Scaled Online Language Learning Application

The advent of modern electronic methods of education has has created opportunities to apply these power-law forgetting models at massive scales. Large online educational programs such as Rosetta Stone®, Khan Academy, and massively open online courses (MOOCs) like Coursera and edX are capable of recording every single observation of student performance in their courses. With these data, is it now possible to understand forgetting in real-world (non-laboratory) settings with many users and observations. Consequently, it is also possible to use predictions of future forgetting to enhance the course experience. For example, by prioritizing study material on the threshold of forgetting (at the *desirable difficulty* suggested by [4]) or by explicitly optimizing retention intervals [12].

0.1.3 Background on The Rosetta Stone® Course

First I will give some background information about the Rosetta Stone® course product from which the data in this thesis are drawn. Next, the specific lessons that are the focus of this study will be summarized. Following this summary, I present a more detailed description of the specific data set used in this thesis. I will also highlight several attributes of the data set that are relevant to this study. Finally, I will describe the specific features collected for each data point which will be used in a later Chapter to fit forgetting models that are individualized to the student.

The data are drawn from the Rosetta Stone® TOTALe Course language learning product. Each TOTALe Course language is composed of one to five language *levels*, which are designed to be taken in series. Each successive level builds on material learned in the previous level. Each level is divided into four *units*, which are in turn subdivided into four *lessons* each. Lessons in a level are designed to make use of and build upon the content of previous lessons. The essential content of the lesson is introduced in an activity labeled as the *core lesson*. Depending on a student's preferences, the student may also be presented with a number of specialty activities. These specialty activities use similar content to that introduced in the core lesson, but focus on particular skills such as vocabulary, pronunciation, grammar, and reading. This hierarchy is illustrated in the entity relationship diagram in Figure 1.

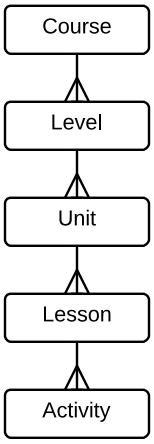

Figure 1: Entity Relationship Diagram showing the organization of the Rosetta Stone® course

Figure 2 shows the Rosetta Stone® home screen. The program will always make a recommendation for what to do next. This might be the next lesson in the curriculum, to review an old lesson, or to schedule a live coaching session. From this screen, the student may navigate to any core lesson or activity in the curriculum (even to a lesson beyond the limit of what he's done so far). The green checkmarks or red cross on each activity indicate whether the learner's score met the predetermined *score threshold* for that activity.

Figure 2: Screenshot example of the Rosetta Stone navigation screen. In this case, the system is making an automated suggestion that the student review the content from Unit 2, Lesson 1.

0.1.3.1 Review Activities

The student is also presented with a special activity called the *review* activity. This review activity is meant to evaluate the student and contains no new material - it simply tests the student on material from previous activities in the same lesson. Each review activity typically consists of eight to ten screens, where each screen presents between two and eight *challenges* to the student. Each challenge represents one discrete interaction the student has with the system. Challenges vary by their prompting media: text, audio, or an image. Each challenge also defines the mode of interaction the student will use to respond to it: clicking an image or text, speaking a response out loud, or typing a free-response answer. There are many permutations of challenge media and response type. For example, suppose the prompt is an audio clip with the spoken words "The woman is running.". In this context, the possible responses could be text, images, or keyboard entry. Each screen's challenges are all of the same type.

Figure 3 shows three different combinations of challenge responses. In the top-left example, the student is prompted to select a picture. In the top-right example, the student is prompted to select from a number of text/audio options. In the bottom example, the student must select from a number of text-only options to fill-in-the-blanks of a sentence, one challenge at a time. The challenges in each screen are presented in a random order. Figure 4 shows a large screen shot of such a review activity.

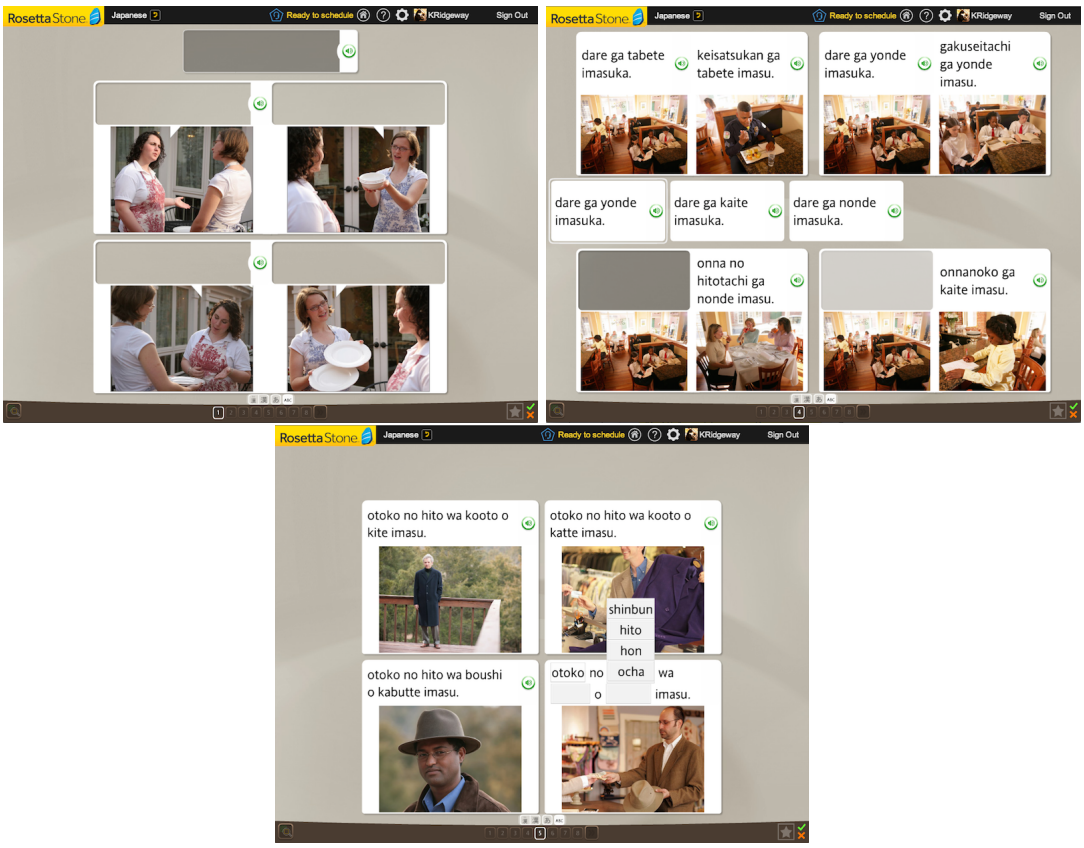

Figure 3: Three examples of different kinds of challenges.

Figure 5 shows the program flow of the screen in Fig. 4. The sequence flows from left to right, and from top to bottom. The first screen, shown at the top-left, plays an audio clip for the student. In this case, the audio is “onna no hito wa tabete imasu” (“The woman is eating”). In the next panel, top-right, the student selects the image of the boys eating. She is given feedback that this was the incorrect choice (a red cross), and the audio is played again. Next (bottom-left),

Figure 4: An example multiple-choice listening challenge screen from a Japanese review activity. First, the prompt sound is played. Next, the student must match the prompt sound with one of the image responses. If the student chooses correctly on his first attempt, that challenge is marked correct. This screen is composed of four such challenges, and this activity has eight screens. After the student completes a challenge, the response options are randomized to discourage obtaining an answer through process of elimination.

she selects the correct image and is given feedback that she was correct (a green checkmark). This sequence represented one challenge of four. Once all the challenges have been answered, the correct answers for all challenges are shown (bottom-right).

Unlike other focused activity types, review activities do not allow the student to skip challenges, to view the correct answers for a screen, or to go back to previous challenges to amend their responses and improve their score. These properties make review activities an excellent opportunity to study retention of material over time, since they should be a good representation of the student's knowledge of the material at that time.

Nonetheless, if the student fails to respond correctly, the system will allow the student to

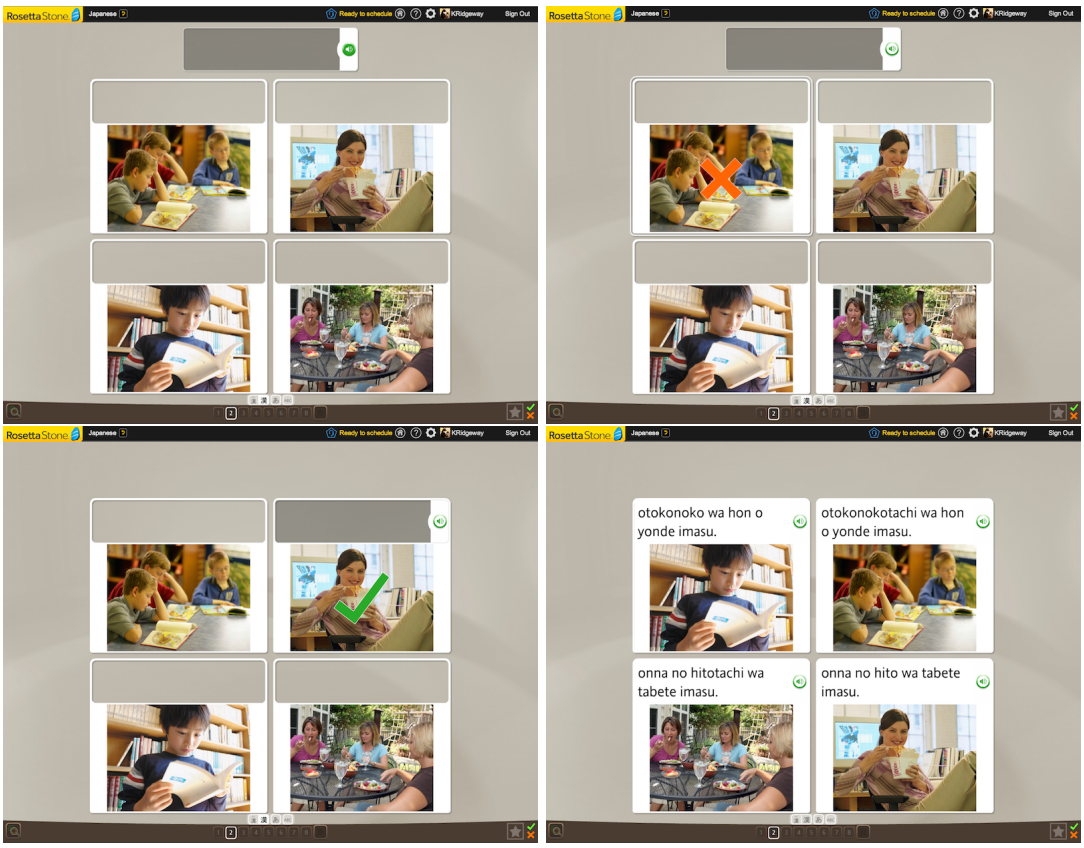

Figure 5: An example sequence of interactions a student can have on the review activity.

continue to respond until he has chosen correctly. If the response mode is speaking, he is given 3 attempts to reach a sufficient score before the activity re-plays the correct native voicing and moves on. However, he is only marked correct if his first response was correct. In this fashion, students are both assessed on their performance on the material as well as given an opportunity to learn material they'd previously forgotten.

The review activities are presented to students at predetermined points in the curriculum. Additionally, the system will periodically bring back old review activities for the student to practice on, based on a simple algorithm (called Adaptive Recall® in the product [10]) designed to schedule activities for review at increased spacing lengths. This system will, as an initial default, suggest that the student review an activity 14 days after the initial attempt. Figure 6 is a screen shot of a message shown to the student after he completes a review activity, informing him that he will be

prompted to repeat this activity after an interval of time (by default, two weeks).

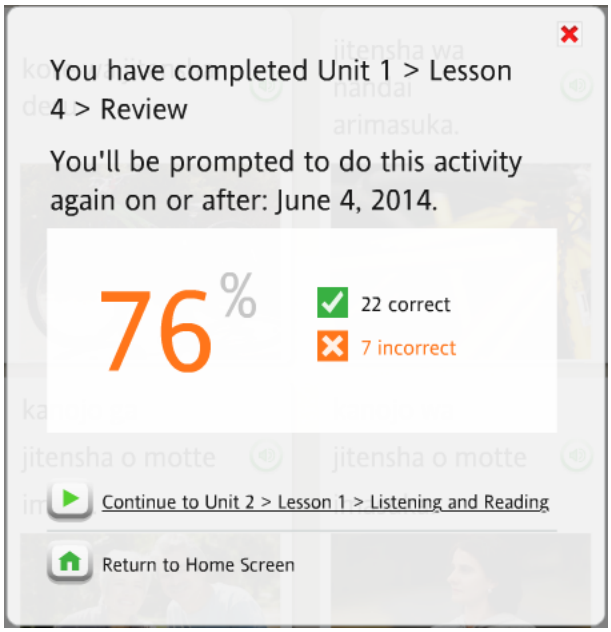

Figure 6: A message shown at the end of a review activity. It gives an indication of when the activity will be rescheduled for review next.

Alternatively, the student can choose to repeat a review activity at any point in the curriculum. Since the same review activity is often repeated multiple times by a student, the difference in scores between successive attempts can be used as a measure of retention for the material in that lesson. Performance on a second attempt will be determined by a combination of factors including learning from the first attempt and forgetting of the material in the intervening time. Additionally, there is strong support for learning taking place even in material designed primarily as an assessment [4]. Therefore, if a student repeats the review activity after a small amount of time, she should show improved performance due to learning. If a student repeats the activity after a large amount of time has passed, her performance should degrade, showing evidence of forgetting the material.

0.1.3.2 Rosetta Stone® Data Set

This work uses a data set with 46.3 million recorded anonymized student activity attempts from the Latin American Spanish products, Levels 1,2, and 3. Since each level is composed of four units, each unit in turn contains four lessons, and each lesson has one review activity, there are a total of 48 review activities in the entire data set. The objective of this study is to predict a single student's second complete attempt at a particular review activity, given the amount of intervening time and other attributes of their learning history. A review activity is marked complete if the student attempts all the challenges in the activity. It is possible for a student to begin a review activity but elect to leave early. These partial attempts are predicted by this model. Information about partial attempts is, however, included in the list of features.

This data set only has information about the aggregated activity scores, and lacks information about how a student responded to individual challenges. Therefore, all of the predictions and features used will be at the activity-level, which combines performance across challenges.

There are relatively few students performing review activities more than twice, so I limited my investigation to predict only the second attempt in order to have enough data points across all activities. The data were further sub-selected to only include students who had completed the core lesson (the lesson that introduces the content tested in the review activity) and who had completed all of the challenges in the review activity on both attempts. In total, there are 545,629 unique student-review activity combinations in the data set that meet the above requirements.

Although the data are anonymized, each review activity observation is marked with a unique student identifier. From this identifier, we can deduce that there are 125,112 unique students in the data set.

The data are also made up of a wide range of intervals of time between attempts on a review activity. The smallest measured intervals are mere seconds long, the longest over 5 years in length. Figure 7 shows a histogram of all time intervals found in the data set.

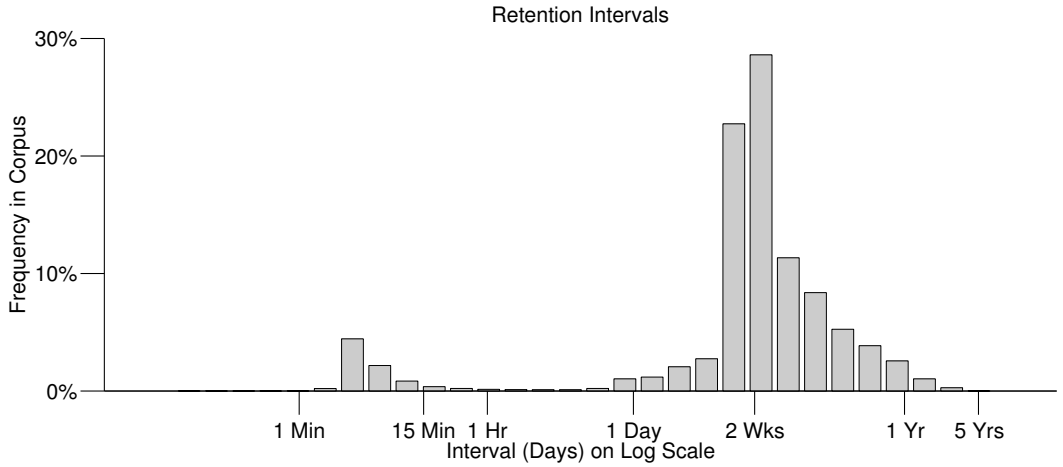

Figure 7: Histogram of retention intervals measured in the data set. This bimodal distribution can be attributed to two aspects of the product. One, the course allows students the freedom to repeat activities at will, to earn a better score. So, after a student completes a review activity, she is free to simply repeat it immediately after completing it to try again. The second mode of the distribution, at roughly 14 days in length is likely attributable to the design of the Adaptive Recall®. This feature will automatically schedule a review activity to be repeated two weeks after the initial attempt. Although the student has the ability to opt-out of the scheduled review, this default suggestion is clearly being followed in the product.

Large online courses are known to have high rates of attrition [18]. If the curriculum of a course is arranged in a sequential order, then it is likely that late-curriculum content will have fewer observed data points to use for fitting and for evaluating models. Indeed, this pattern is observable in the Rosetta Stone data set, visualized in Figure 8.

This trend has implications for how I evaluate the models built in this thesis. If each student observation is weighted equally, then the activities early in the curriculum will be over-represented, and the late activities with few data points will be under-represented. One of the goals of this work is to create a model that can predict well across all activities **and** for all students. Therefore, this thesis will focus on reporting activity-weighted performance results. I will also report student-weighted results for comparison.

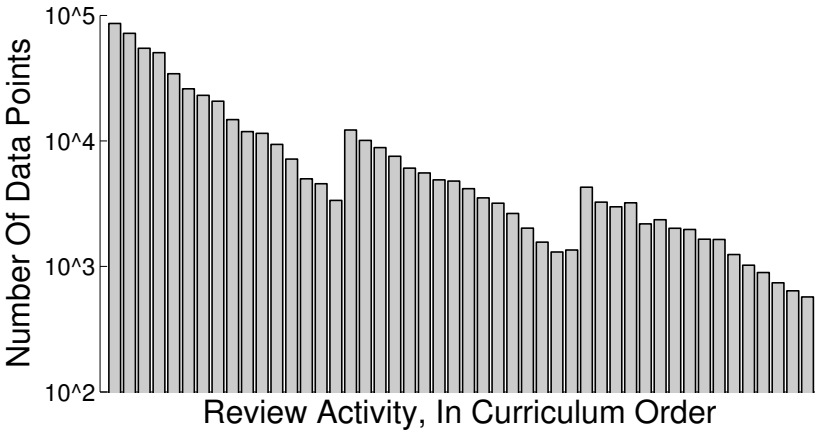

Figure 8: Data points per activity, for all 48 activities in the data set. Each bar represents one activity. Its height indicates how many data points are in the data set for that activity. To help visualize differences in number of data points, that axis is plotted with a log scale. The two bumps in this graph represent the first lesson of a level. Since the product is sold by level, and learners typically begin at the first lesson of the curriculum, these bumps represent the addition of new learners.

0.1.3.3 Features of the Spanish Data Set

For each student in each review activity, I extracted twenty distinct features from the database which seemed to be potentially useful predictors of student performance. Each feature is prefixed with a short name that will be used to refer back to it in later Chapters. Scores are all proportions correct in the range $[0, 1]$. Activity times are all in seconds, unless indicated otherwise, and are in the range $(0, +\infty)$.

(1) Information about a student's performance on the review activity.

Score1 The student's score on their first attempt at the review activity, hereafter, s_1. This is a fraction representing the number of challenges in the review activity the student answered correctly on the first attempt divided by the total number of challenges attempted.

DeltaTime The intervening time, in days, between the first attempt and the second attempt of the review activity.

TimeSpent The amount of time spent on the first attempt of the review activity

(2) **SecondIsAR** A Binary variable (0 or 1) indicating whether the second attempt is scheduled with Adaptive Recall®.

(3) Information about a student's performance on the core lesson. As discussed in Section 0.1.3, core lessons introduce the content tested by the review activities.

CoreScore The student's score on the core lesson. As discussed in Section 0.1.3, core lessons introduce the content tested by the review activities.

CoreTime The amount of time, in seconds, the student spent completing the core lesson.

(4) Information about incomplete (partial) attempts of this activity in the intervening time between the two completed review activity attempts.

PartialTime The amount of time, in seconds, the student spent in partials.

PartialCount The number of such incomplete attempts in the intervening time.

(5) Information about other activity types within the same lesson that were completed before their second attempt.

LessonActivityScoreStd The standard deviation of the scores of these activities

LessonActivityTypeCount How many types of activities in the lesson were completed

LessonActivityTime The amount of time spent in these activities

LessonActivityCount How many individual activity attempts were made

(6) Binary indicator variables (0 or 1) that are set to 1 if they completed a certain activity type within the same lesson as the review activity, before attempting the review activity a second time.

DidWriting

DidGrammar

DidListening

DidListeningAndReading

DidSpeaking

DidPronunciation

DidVocabulary

DidReading

0.2 Methodology

One of the goals of this work is to compare the performance of various models on the review activities in the data set. In general, the methods used to train and test the models need to show that the models are highly generalizable, make use of metrics that are comparable across models and review activities, and deliver results that are intuitively interpretable.

The method for aggregating test error on data points is an important consideration in this work. The most obvious option is to weight each observation equally. However, given the wide disparity in data set sizes across activities, this appraoch may over-represent very popular review activities. Another option is for each activity to carry equal weight. This would be useful for the product, since it is beneficial to have a model that predicts well across a large spread of activities. This work focuses on balancing performance across activities (activity-weighted error), but also reports per-data point weighted error (student-weighted error) when appropriate.

The following training and evaluation procedure has been devised in support of these goals.

0.2.1 Cross Validation Training and Test Procedure

Consider a model M with a set of free parameters θ, a vector of outcomes y, a matrix of features X, and an error function that compares predictions $\hat{y}$ with actual values y, $\text{err}(\hat{y}, y)$. The goal of this training and test procedure can be expressed as a search for the settings of θ to minimize err for training outcomes y_{train} given training data X_{train} and the set of all values of theta Θ:

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \text{err}(M(X_{\text{train}}, \theta), y_{\text{train}}) \tag{3}$$

In order to evaluate the performance for the estimated $\hat{\theta}$, some held-out test data y_{test} and X_{test} are used:

$$\text{err}(M(X_{\text{test}}, \hat{\theta}), y_{\text{test}}) \tag{4}$$

Therefore, in order to ensure that the estimated model is able to generalize well to unseen data, it is necessary to split the data set into a training set and a test set. However, a single choice

for such a split can be biased in some way. It is often useful to employ a cross validation procedure to avoid such situations. In cross validation, the training/test procedure is repeated n times such that the union of all the test sets used cover the entire data set. For each cross-validation split, the training set is used to fit a separate model θ. The model is then used to predict the test set of that split. The error metric reported is equal to the error function applied to the union of all data points in all test sets. Each review activity is considered a data set and has its own set of cross-validation splits.

0.2.2 Normalized Error Metric

An appropriate error metric is needed to evaluate the performance of the model. Simply reporting the sum squared residuals is inappropriate because different activities have different amounts of data and, thus, different numbers of testing points. Also, an intuitively interpretable metric should compare the performance of the model to some reasonable baseline. For these data, a reasonable baseline is to predict the mean of the training set for all points in a test set. The error metric used in this work is, therefore, the sum of the squared residuals, normalized by the residuals of predicting the mean of the training set. More formally,

$$\mathrm{err}_{\mathrm{norm}}(y, \hat{y}) = \frac{\Sigma_{i=1}^{N}(y_i - \min(1, \max(\hat{y_i}, 0)))^2}{\Sigma_{i=1}^{N}(y_i - \bar{y}_{\mathrm{train}})^2} \tag{5}$$

where N is the number of data points in y and $\hat{y}$, y_i is the actual outcome for data point i, $\hat{y_i}$ is the model-predicted value for data point i, and $\bar{y}_{\mathrm{train}}$ is the mean over outcomes y_{train} of the training set. The model-predicted outcomes are limited to the range $[0, 1]$ because they are proportions correct in this work. If the $\mathrm{err}_{\mathrm{norm}}$ term is less than 1.0, then the model's predictions are better than predicting the mean of the training set.

0.2.2.1 Interpretation Of Normalized Error Metric As Percent of Variance Unexplained

The error metric in Eq. 5 is related to the coefficient of determination. Its formula is

$$\mathrm{R}^2 = 1 - \frac{\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2}{\Sigma_{i=1}^{N}(y_i - \bar{y})^2} \tag{6}$$

The law of large numbers says that as the data sets grow in size, $\bar{y}_{\text{train}}$ should converge to $\bar{y}$. In the limit of N$\rightarrow \infty$, Eq. 6 is equal to $1 - \text{err}_{\text{norm}}$. Since the coefficient of determination is commonly interpreted as the percent of variance explained, the normalized error metric in Eq. 5 can be interpreted to be the percent of variance in the test set unexplained by the model given the training set.

0.2.3 Comparison Between Models

To compare two models, they must both be evaluated on each of review activities using the procedure in Section. 0.2.1 and the error metric in Section 0.2.2.

To compare two models' performance across activities, I compute, for each model and activity, the mean error across test splits. A two-tailed, paired-sample t test, where each pair represents activity performance for each model, is performed and reported to test for significance. Finally, the mean activity-weighted error and the student-weighted error are reported for both models.

0.2.4 Nonlinear Fitting Procedure For Power-Law Models

Fitting the three-parameter power law model in Eq. 1 from Chapter 0.1 to the data requires nonlinear optimization. The three-parameter power law model built in this work is trained using MATLAB's `nlinfit` function. For each cross validation split, the fit is performed 15 times using initial values drawn randomly. I evaluate the test set with the best-fitting model (the model with the lowest sum of squared residuals over the training set). Initial values for α are drawn from a uniform distribution over $[0.9, 1.0]$, the time-scaling values γ are drawn from a uniform distribution over $[0, 10^5]$, and β values are drawn from a uniform distribution over $[-.01, 0]$. These β values are drawn from a relatively narrow range. I determined these distributions empirically through experimentation. Less constrained initial β values made `nlinfit` emit errors.

0.2.4.1 Comparing Nonlinear and Linear Fitting

The two-parameter power law model from Chapter 0.1 in Eq. 2 can be fit in two ways: by using the procedure from Section 0.2.4 or by using least-squares linear regression to optimize an equivalent model in the logarithmic domain.

$$\log Pr(\text{recall}) = \log \alpha + \beta \log t \tag{7}$$

Note that this linear method minimizes the squared error on the logarithm of the prediction, rather than on the raw prediction. Therefore, one cannot assume that the two training procedures will produce the same results.

To compare the two fitting procedures, I estimated the parameters of the two-parameter power law function in Eq. 10 using both MATLAB's `nlinfit` function and the least-squares procedure described in Eq. 7 using 5-fold cross-validation. Errors for each model are computed on the held-out test sets. If the estimated models are significantly different, this should be reflected in their error.

Figure 9 shows the differences in activity error for least squares and `nlinfit` fits. Each bar represents one of the 48 review activities in the data set. The height of the bar indicates the difference in normalized error between the `nlinfit`-fit model and the least-squares-fit model. A negative difference indicates that the `nlinfit` model had lower error. The `nlinfit` model has a slightly lower activity-weighted error: 0.8858 vs 0.8863 ($t(47) = -35.8648, p < 0.05$ with a paired-sample, two-tailed t-test).

Figure 9: Differences between the test errors produced by the two fitting procedures.

However, when inspecting the activity-weighted differences in error in Fig. 9, it is clear that most of the differences are very small, with a mean difference in activity error of only 0.0005, which is only a 0.06% gain relative to the nonlinear activity-weighted error. There does appear to be a statistically significant advantage to nonlinear fitting. However, due to the small differences, I conclude that the least-squares fitting procedure is a reasonable substitute for the nonlinear one.

0.3 Models of Forgetting

Before delving into investigations and comparisons of individual models, I will give an overview of the models explored in this chapter. These models and their relationships are summarized in Figure 10. It may be useful to the reader to refer back to this figure as they move through the chapter. I have also color coded the model names to assist the reader in keeping them distinct.

How well can power-law forgetting models such as the three-parameter model in Eq. 1 and the two-parameter model in Eq. 2 characterize the real-world Rosetta Stone review path data sets? How robust are the models across activities? To address questions such as these, both power-law forgetting prediction models will be fitted and evaluated using the procedure in Chapter 0.2. In this work, the three-parameter power law forgetting model is referred to as $\mathrm{PL}_{\alpha,\beta,\gamma}$, two-parameter power-law model as $\mathrm{PL}_{\alpha,\beta}$.

Next, I will investigate the effect of including individualized features as part of the prediction model. The two- and three-parameter power-law forgetting models are characteristic of an approach typical of psychology studies, whose goal is to robustly fit data from a large population of individuals studying some set of material. In contrast, a generic statistical approach is to predict scores for individual students using an array of features that are specific to that student: in this case, information about their specific study history. To begin, a linear regression model, Linear(f), is built that predicts a student's performance as a linear combination of the relevant features of their study history. The Linear(f) model is then fitted to evaluated on the Rosetta Stone data, and compared with the $\mathrm{PL}_{\alpha,\beta}$ model. The Linear(f) model is shown to outperform $\mathrm{PL}_{\alpha,\beta}$, but to have a tendency to overfit review activities with small numbers of data points.

Finally, a hybrid model will be developed that combines the power-law forgetting model and the individualized prediction model by replacing the alpha and/or beta parameters of the power law model with linear functions of the features, yielding models in which alpha is feature dependent ($\mathrm{PL}_{\alpha(\mathrm{f}),\beta}$), beta is feature dependent ($\mathrm{PL}_{\alpha,\beta(\mathrm{f})}$) and both alpha and beta are feature dependent ($\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$). The final combined model, with both power-law coefficients estimated via

a combination of features ($\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$), will be shown to outperform the linear regression model, the power-law forgetting models, and both intermediate combinations. Figure. 10 shows a visual representation of the relationships between these models. Both the $\text{Linear}(\text{f})$and $\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$models will then be extended with second-order features ($\text{Linear}(\text{f},\text{f}^2)$and $\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$), but will be shown to overfit the smaller review activities. Chapter 0.4 shows how to apply regularized regression to prevent overfitting of these smaller activities, and presents a detailed analysis of the individualized features that contribute to predictions of learner performance.

0.3.1 Power-Law Forgetting

0.3.1.1 Three-Parameter Power-Law Forgetting

The three-parameter power-law forgetting curve in Eq. 1 relates the probability of recall, $Pr(\text{recall})$, to the time value t:

$$Pr(\text{recall}) = \alpha(1+\gamma t)^{\beta} \tag{8}$$

In the context of this work, $Pr(\text{recall})$, the probability of recall, is assumed to be linearly related to score on the second attempt at the review activity (s_2) after a time interval t, in days (Eq. 9). In the forgetting curve in Eq. 8, α is a value in the range $[0,1]$, indicating the probability of initial learning (or, the probability of recall at $t=0$). The second score of the review activity s_2 is the fraction of challenges answered correctly and, like the probability of recall, lies in the range $[0,1]$. Therefore, when predicting s_2 instead of $Pr(\text{recall})$, the α value should also be in the range $[0,1]$. The β parameter is the rate of forgetting and lies in the range $(-\infty,0)$. The γ parameter is a scaling factor on time, and lies in the range $(0,+\infty)$. This model, in Eq. 9 is referred to as $\text{PL}_{\alpha,\beta,\gamma}$.

$$s_2 = \alpha(1+\gamma t)^{\beta} \tag{9}$$

As a first step in understanding the factors affecting student performance in these data, I

investigate the role that forgetting plays. To what extent can the observed review activity scores be characterized by a power-law forgetting function of the intervening time? To answer this question, I fit the three-parameter model to the data in each review activity.

I use the training and cross-validation evaluation procedure from Chapter 0.2. The features consist of the time interval in days between review activity attempts, the outcome variable is the second score s_2, and the model θ consists of the coefficients $\{\alpha, \gamma, \beta\}$ to the three-Parameter forgetting model in Eq. 9. Figure. 11 shows normalized error across all 48 activities in the data set.

Recall that an err_{norm} term less than 1.0, means the model's predictions are better than predicting the mean s_2 of the training set, and is related to the percent of variance unexplained by the model. The forgetting curve in Eq. 9 explains a substantial amount of the variance in the data sets. In the best case, the simple population-fitted forgetting $\text{PL}_{\alpha,\beta,\gamma}$accounts for 30% of the variance in a review activity. In the worst case, it only accounts for only 1.3% of the variability. In general, the larger data sets have less of their variance explained by forgetting. Therefore it is not surprising that the overall normalized student-weighted error of these data is 0.9373. The activity-weighted error is is 0.8855. According to the formula in Chapter 0.2, the mean activity-weighted variance explained is $100(1 - 0.8855)$ or 11.5%.

0.3.1.2 Two-Parameter Power-Law Forgetting

The three-parameter model $\text{PL}_{\alpha,\beta,\gamma}$(Eq. 9) explains a significant amount of variance for many review activities. Recall that, according to [17], an alternative formulation of the power-law forgetting curve is a simpler two-parameter power-law, derived by removing the γ term and the unit offset. This model is referred to as $\text{PL}_{\alpha,\beta}$.

$$s_2 = \alpha t^{\beta} \tag{10}$$

Note that, if t or γ become very large, Eq. 9 and Eq. 10 are equivalent since the unit offset term of $(1 + \gamma t)$ will become miniscule relative to γt.

0.3.1.3 Comparison of Two- and Three-Parameter Power-Law Forgetting

It is not certain that a two-parameter model has the same expressive power as the three-parameter model for these data. As noted in [17], the two-parameter model has been noted to characterize individual forgetting, whereas Wickelgren's three-parameter power law (Eq. 9) is accurate at characterizing population data. Additionally, the three-parameter model is defined at $t = 0$, but the two-parameter model is not. This gives a nice theoretical justification for the three-parameter model, since it directly estimates the amount of learning in the α parameter.

What is the effect of removing the unit offset and γ scaling parameter on t of the three-parameter model? To answer this question, the same cross validation training and test procedure used to evaluate the three-parameter model is also applied to the two-parameter model in Eq. 10.

Note that, since there will be many significance tests performed in this chapter, it is important to choose a significance level reflective of that fact. A common method for addressing this issue is to use a Bonferroni correction on the p-value necessary for significance. We chose a p-value (.0025) to have a 5% chance of error over the 10 two-tailed t tests performed in this chapter.

The mean activity-weighted error is 0.8855 for $\mathrm{PL}_{\alpha,\beta,\gamma}$vs. 0.8858 for $\mathrm{PL}_{\alpha,\beta}(t(47) = 1.2724, p = 0.2095)$. The student-weighted error for $\mathrm{PL}_{\alpha,\beta,\gamma}$ is 0.9373 vs. 0.9378 for $\mathrm{PL}_{\alpha,\beta}$. The individual differences in activity error can be seen in Fig. 12.

The actual differences in error are quite small, as seen in in Fig. 12. The largest difference in error for any activity is 0.01, about 1% relative to either model, and most are much smaller. Also, there is not obvious advantage to one model over the other. Given the small differences in absolute error, it is reasonable to conclude that the two-parameter model has similar power to describe forgetting in these data as the three-parameter model.

The mean activity-weighted differences between the two models can be seen in Fig. 13, where $\mathrm{PL}_{\alpha,\beta,\gamma}$ and $\mathrm{PL}_{\alpha,\beta}$ represent the first two bars, respectively. All of the results in the rest of this chapter will refer back to Fig. 13.

0.3.1.4 Visualization of Two-Parameter Power-Law Forgetting

To better understand the variability in prediction error across activities (Fig. 11), it can be useful to visualize the curve fits for individual review activities. To this end, s_2 can be plotted as a function of the intervening time t. When graphed in log-log coordinates, a power function is expressed as a straight line. The visualizations in Figures 14, 15, 16, and 17 plot both the fitted functions, in red, and points representing the observations in the data set, in blue.

To obtain a cleaner-looking plot, the data points are binned by their time intervals such that each bin has exactly 50 data points. The plotted points represent the mean time interval and the mean s_2 for each bin. This 50-fold reduction makes the data set manageable for the purpose of visualization. The power law functions in red are fitted to the underlying data points, not the binned points.

Figures 14 and 15 show this visualization for the best- and worst-fitting 3 review activities for the $\mathrm{PL}_{\alpha,\beta}$ model, as determined by the normalized error metric. Figs. 16 and 17 represent the review activites with the smallest and largest number of data points, respectively.

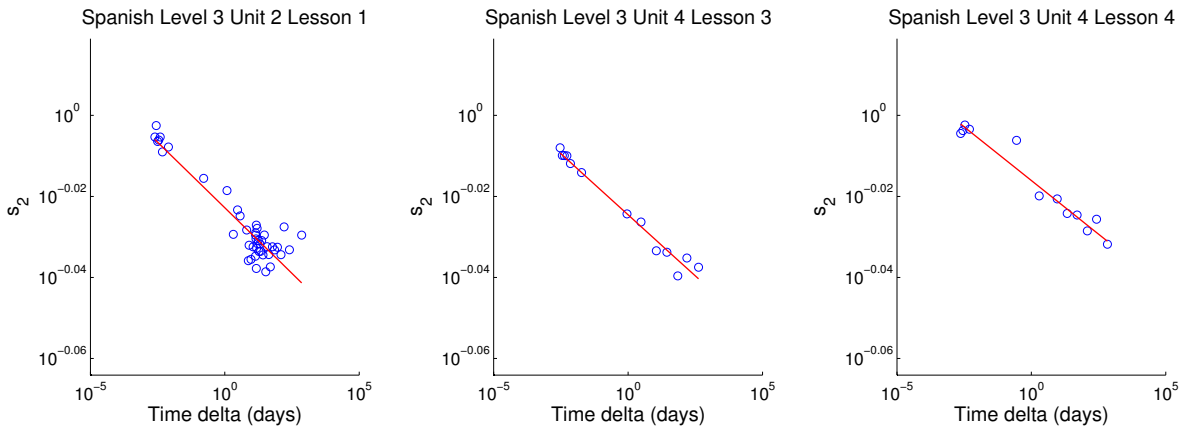

Figure 14: Two-Parameter forgetting fits for the best-fitting 3 review activities, according to the normalized error in Fig. 11. The red line is the power-law function, which shows up as a linear relationship in this logarithmically scaled plot. The blue circles each represent the mean time and s_2 of one bin of 50 data points. The red power-law function is fitted to the underlying individual data points.

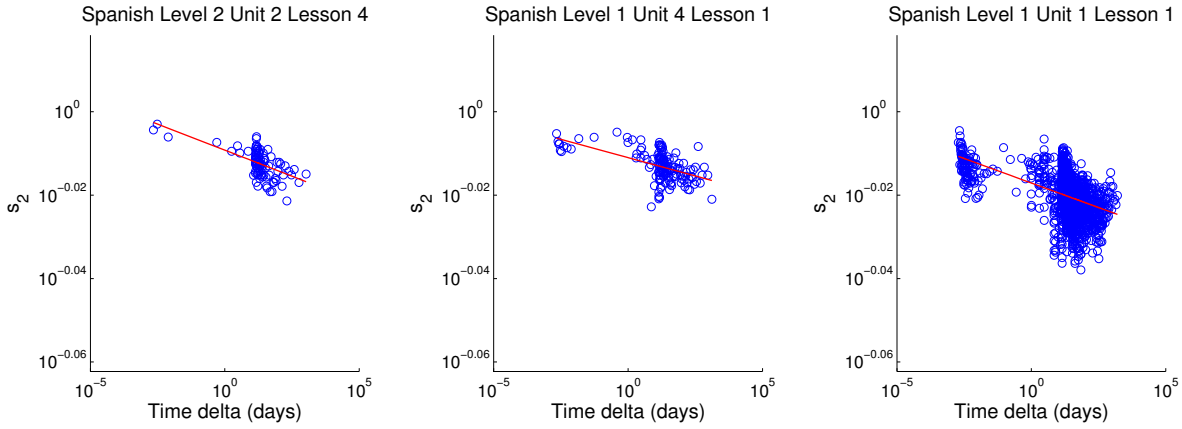

Figure 15: Two-Parameter forgetting fits for the worst-fitting 3 review activities, according to the normalized error in Fig. 11.

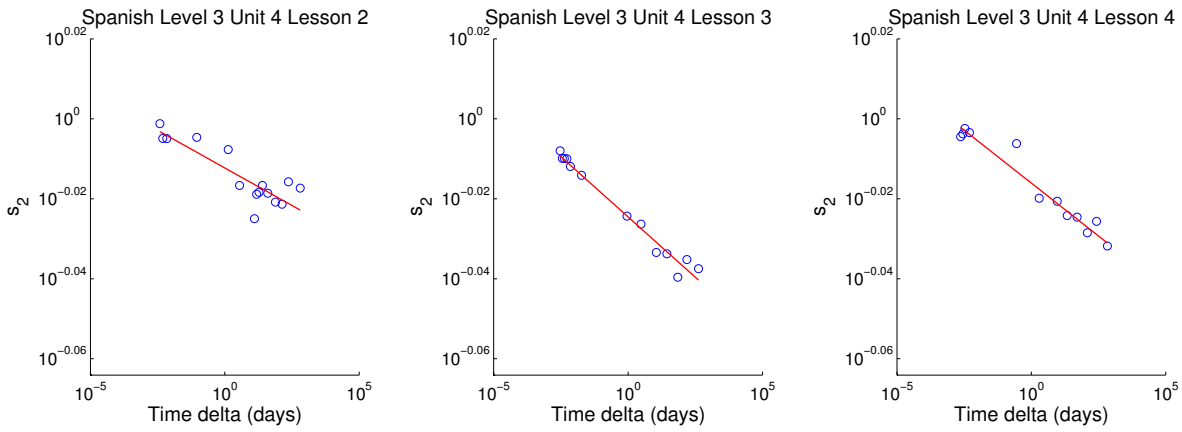

Figure 16: Two-Parameter forgetting fits for the 3 review activities with the smallest number of data points. From left to right, these sets have 741, 640, and 571 student observations.

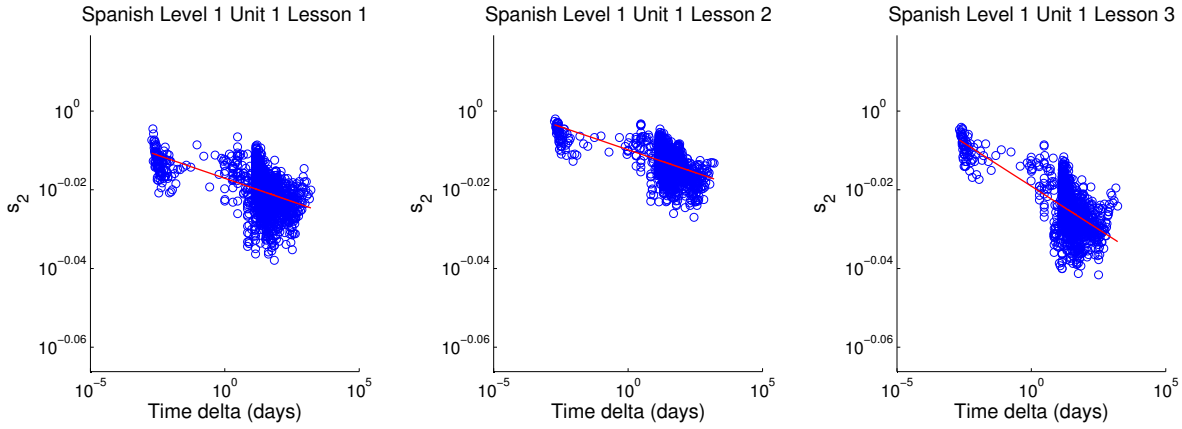

Figure 17: Two-Parameter forgetting fits for the 3 review activities with the largest number of data points. From left to right, these sets have 86293, 72025, and 54709 student observations. The large groups of data at the 14-day interval are due to the default review scheduling policy of Adaptive Recall®, as mentioned in Chapter 0.1 Section 0.1.3.1.

0.3.1.5 Variance In Early Activities

Some activities seem to exhibit less forgetting than others. To illustrate this, I compare one of the worst-fitted paths to one of the best-fitted to investigate this potential issue.

According to Fig. 18, there is more forgetting, on average, in Level 3 Unit 2 Lesson 1 than in Level 2 Unit 2 Lesson 4. Is this simply a function of which learners are engaged in which activity? Consider Fig. 19, which shows both activities, but only with the subset of learners who completed both activities.

The two forgetting curves for Level 2, Unit 2, Lesson 4, containing different populations of learners, are almost identical. Similarly the curves for Level 3, Unit 2, Lesson 1 are also very similar. If the inter-activity variance is not explained by user identity, then it is possible that different activities show different amounts of evidence of forgetting. The example activities in Fig. 18 seem to suggest that this might be the case. As noted in Chapter 0.1.3, each lesson within a level is designed to use and build upon material taught in previous lessons. If this is true, then the content in each lesson will act to reinforce content from earlier lessons within the level. For a lesson early in the curriculum like L2-U2-L4, the learner will be presented with more activities that touch upon the content of that lesson in the time between review attempts than for a later lesson like L3-U2-L1. Therefore, forgetting should be less strong for content in early lessons, and stronger for content in later lessons.

However, there is no clear pattern of increasing forgetting for lessons later in the curriculum than for lessons earlier in the curriculum. Fig. 20 shows the β coefficients for all activities, sorted by their position in the curriculum, by language level. A lower (more negative) value of β indicates faster forgetting.

Although the first lessons tend to be forgotten faster than the last lessons in each level, there is no strong pattern of forgetting discernible across activities in the curriculum. However, there is clearly a range of forgetting across activities. To shed light on the variance in forgetting across activities, it would help to conduct a further analysis that takes into account exactly what content

is shared between lessons, and which exact activities were done by learners in the intervening times. However, this exploration is beyond the scope of this thesis.

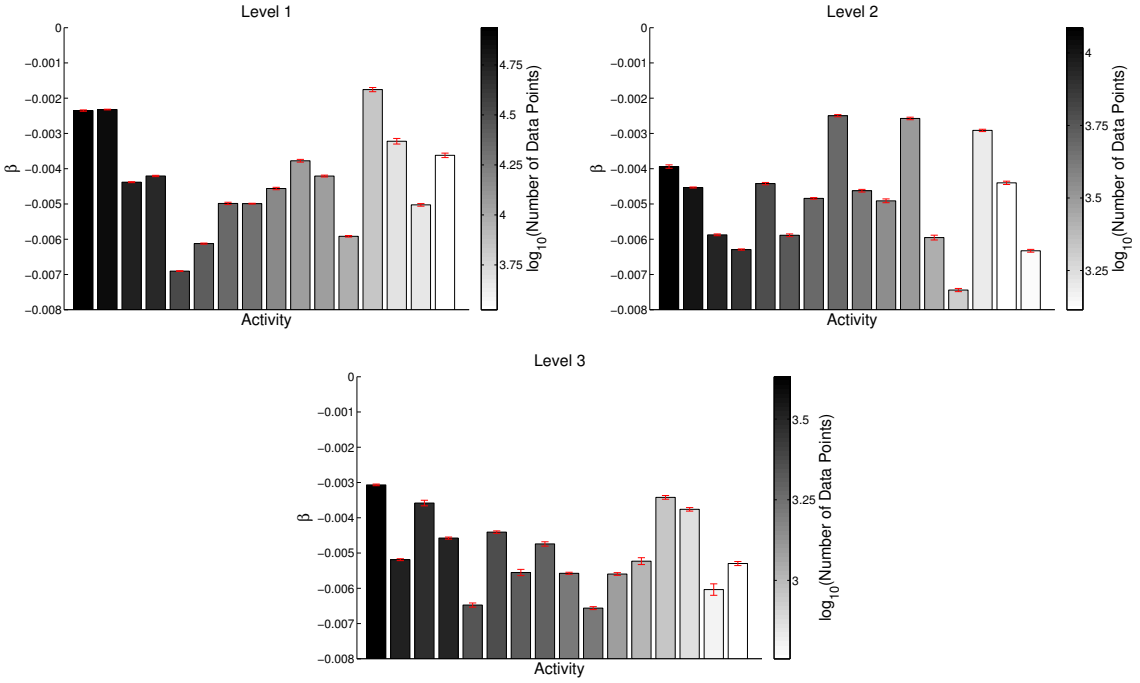

Figure 20: Mean β coefficient values per activity, broken down by level and presented in the order in which they are introduced in the curriculum. Error bars represent +/- 1 standard error of β between cross validation splits.

0.3.2 Linear Regression

Having discussed power-law forgetting fits to the data, I now investigate the second class of models I considered to predict student performance: Linear regression. Linear regression is typically the first model one fits to data in statistics and machine learning. In this case, I use linear regression to predict student performance as a function of various attributes of the student's study history. A linear regression is fitted to predict s_2 as a function of a set of N regression features $X_0 ... X_N$ multiplied respectively by a set of coefficients $b_0 ... b_N$. In this case, I use the 20 features described in detail in Section 0.1.3.3. Features such as the learner's score on her first attempt, her

score on the core lesson, and how much time she spent on any intervening lessons are, of course, individualized to the learner. Therefore, this model makes predictions that are specific to this learner, rather than just specific to the activity.

$$s_2 = \sum_i^N b_i X_i \tag{11}$$

The Linear(f) model activity-weighted error is 0.8001 vs. 0.8858 for the $\text{PL}_{\alpha,\beta}$ model ($t(47) = -8.8404, p < 0.05$ by a two-tailed Bonferroni corrected t-test, with the activity as the random variable. All significance tests in this Chapter are performed at this significance level.). The per-data point results also showed an improvement 0.7929 for Linear(f) vs. 0.9378 for $\text{PL}_{\alpha,\beta}$. The activity prediction errors are shown in Fig. 21.

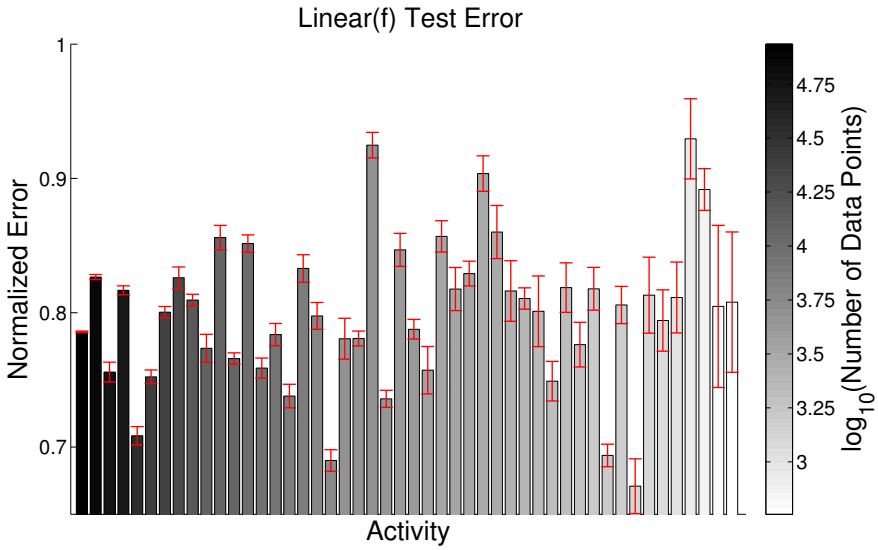

Figure 21: Activity error for the Linear(f) model in Eq. 11. Error bars represent standard error from the cross validation splits.

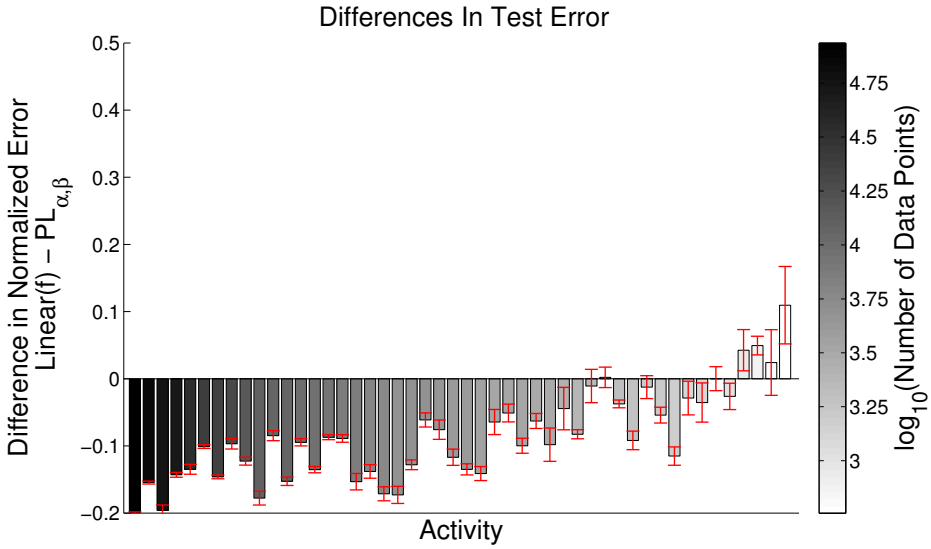

Figure 22: Differences in error between linear regression and power-law forgetting. Each bar represents the activity error on Linear(f) minus the error on $\text{PL}_{\alpha,\beta}$. Negative numbers indicate an advantage for Linear(f), positive numbers an advantage for $\text{PL}_{\alpha,\beta}$. The activities are sorted by number of data points. Error bars represent +/- 1 standard error from cross-validation splits.

The differences in error between the Linear(f) and $\text{PL}_{\alpha,\beta}$ models reveals an intriguing pattern (Fig. 22). The differences are mostly negative, reflecting the lower prediction error of Linear(f) over $\text{PL}_{\alpha,\beta}$. However, the activities with relatively few data points are fit better by the simpler two-parameter power law model. This is likely due to overfitting: For example, the last path has only 571 data points and the Linear(f) model has 21 free parameters. Unsurprisingly, activities with many data points are fit better by a more complex model, and activities with few data points are fit better by a model with fewer free parameters.

0.3.3 Combining Power-Law Forgetting And Linear Regression

The two-parameter power-law forgetting model $\text{PL}_{\alpha,\beta}$ in Eq. 10 makes it possible to recast the power-law model fitting problem as linear regression, but in the log domain, predicting $\log s_2$.

$$\log s_2 = \alpha' + \beta \log t \tag{12}$$

Eq. 12 is linear in $\log s_2$ with the feature $\log t$ and the constant offset term α'. Note that α' is the log of the original α term. This is equivalent to the nonlinear two-parameter forgetting model:

$$s_2 = \exp(\alpha')t^{\beta} \tag{13}$$

Because the power-law and linear regression models can both be expressed as linear regressions, it seems natural to consider a hybrid model that unifies and combines the two models. This single model can be interpreted as a power law model where both α' and/or β are not constants, but are linear functions of features. This model is called $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$.

$$s_2 = \alpha(\mathrm{f})t^{\beta(\mathrm{f})} \tag{14}$$

where $\alpha(\mathrm{f})$ and $\beta(\mathrm{f})$ are defined as:

$$\alpha(\mathrm{f}) = \exp(\sum_{i}^{N} \alpha'_i X_i) \tag{15}$$

$$\beta(\mathrm{f}) = \sum_{i}^{N} \beta_i X_i \tag{16}$$

The single combined model can also be interpreted as a linear model in which the regressand is $\log s_2$ and the regressors include terms multiplied by $\log t$.

$$\log s_2 = \sum_{i}^{N} \alpha'_i X_i + \sum_{i}^{N} \beta_i X_i \log t \tag{17}$$

To understand the contribution of each component of the combined model, two models that are subsets of this combined model will be investigated: one that replaces only β with $\beta(\mathrm{f})$ ($\mathrm{PL}_{\alpha,\beta(\mathrm{f})}$) and one that replaces only α with $\alpha(\mathrm{f})$ ($\mathrm{PL}_{\alpha(\mathrm{f}),\beta}$).

First, I evaluate $\mathrm{PL}_{\alpha,\beta(\mathrm{f})}$:

$$s_2 = \alpha t^{\beta(\mathrm{f})} \tag{18}$$

This model outperforms $\mathrm{PL}_{\alpha,\beta}$ in all measures. Its mean activity-weighted error is 0.8242 vs. 0.8858 for $\mathrm{PL}_{\alpha,\beta}(t(47) = -11.4868, p < .05)$. Its student-weighted error is 0.8356 vs 0.9378 for

$PL_{\alpha,\beta}$.

Next, I evaluate $PL_{\alpha(f),\beta}$:

$$s_2 = \alpha(f)t^{\beta} \tag{19}$$

This model also outperforms $PL_{\alpha,\beta}$. Its mean activity-weighted error is 0.7994 vs 0.8858 for $PL_{\alpha,\beta}(t(47) = -8.9458, p < .05)$. It has a student-weighted error of 0.7916 vs 0.9378 for $PL_{\alpha,\beta}$.

Finally, I evaluate the full combined model $PL_{\alpha(f),\beta(f)}$. This model has an activity-weighted error of 0.7581, and a student-weighted error of 0.7734, making it the best model considered for predicting student performance. T tests comparing this model with $PL_{\alpha,\beta}(t(47) = -17.7871, p < 0.05)$, $Linear(f)(t(47) = -10.4752, p < 0.05)$, $PL_{\alpha,\beta(f)}(t(47) = -15.9858, p < 0.05)$, and $PL_{\alpha(f),\beta}(t(47) = -10.3356, p < 0.05)$ are all significant. Note that this model still lacks the purely additive terms in Eq. 11. These could be incorporated by building a hybrid model, using the additive terms to predict the residuals of Eq. 14.

Figure. 23 shows the activity error for $PL_{\alpha(f),\beta(f)}$. Figure. 24 shows the differences in error between $Linear(f)$ and $PL_{\alpha(f),\beta(f)}$ activity error. All activities had lower error for $PL_{\alpha(f),\beta(f)}$ than for either $Linear(f)$ or $PL_{\alpha,\beta}$.

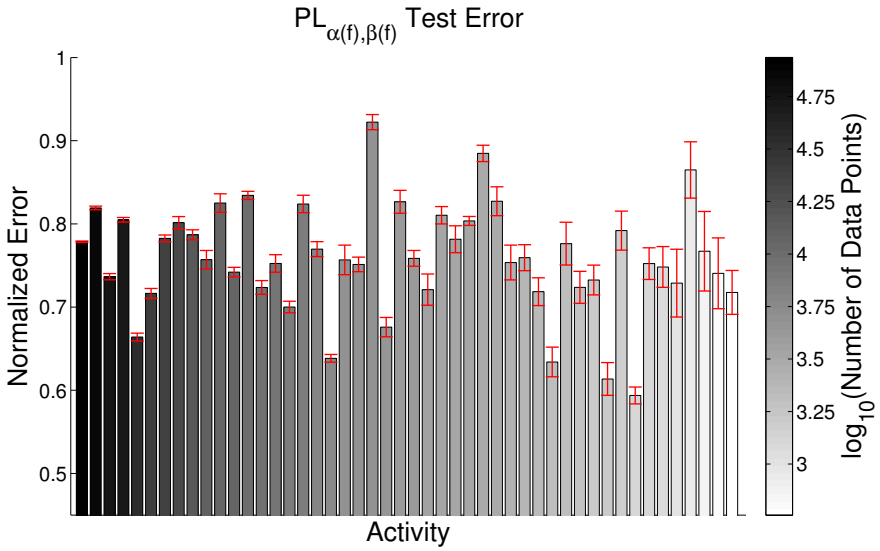

Figure 23: Error for power-law forgetting linear regression model in Eq. 14, replacing α and β with functions composed of linear combinations of student features.

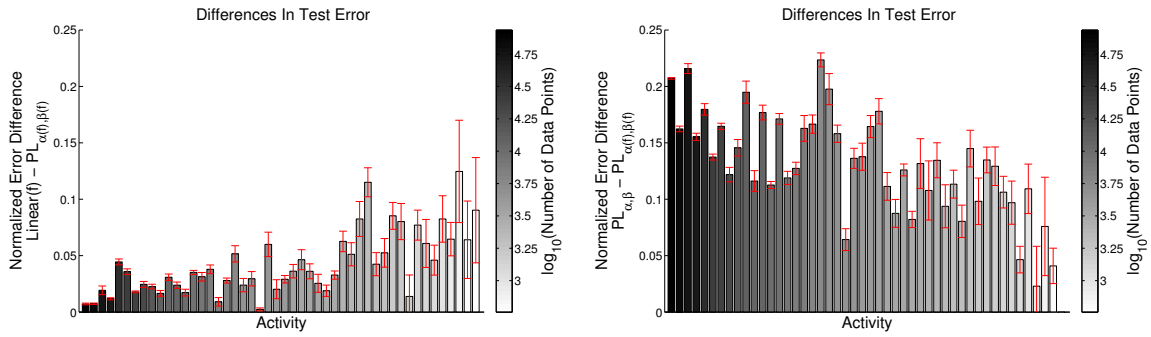

Figure 24: Differences in activity error between Linear(f) and $\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$, and between $\text{PL}_{\alpha,\beta}$ and $\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$. A positive difference in either plot indicates that $\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$ had lower prediction error.

0.3.3.1 Adding Second-Order Features

In multiple linear regression, it is common to add higher-order terms, e.g. a second-order model might include feature values squared and feature value cross-product terms to fit quadratic lines. This is sensible in the context of the power-law function, with terms estimating α and β. Both Linear(f)and $\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$models were augmented with second-order terms and re-tested. For

speed reasons, the set of 8 binary features that mark a learner's attempt on a particular type of activity (e.g. DidWriting, DidListeningAndReading) were omitted. These augmented linear- and power-law models are called Linear(f, f^2)and PL$_{\alpha(\text{f,f}^2),\beta(\text{f,f}^2)}$, respectively.

The linear regression model with second-order terms (Linear(f, f^2)) had an activity-weighted error of 0.8404 vs 0.8001 for the Linear(f)model ($t(47) = 2.2297, p < 0.05$). The addition of so many features (20 raw features, plus 11 squared features omitting all indicators, plus $\frac{12!}{2!(12-2)!} = 66$ cross-product features and a bias term makes 98 total features), unsurprisingly results in an overfit model on the smaller data sets (the smallest has only 571 total data points). However, the squared features did give it a small gain in student-weighted error: 0.7763 vs 0.7929 for Linear(f).

The augmented power law model PL$_{\alpha(\text{f,f}^2),\beta(\text{f,f}^2)}$ had an activity-weighted error of 0.9686 vs 0.7581 for PL$_{\alpha,\beta}$($t(47) = 3.2868, p < 0.05$). This overfitting is unsurprising: the PL$_{\alpha(\text{f,f}^2),\beta(\text{f,f}^2)}$ has the same 98 features of Linear(f, f^2), plus 97 feature values times $\log(t)$, for a total of 195 features.

Regularization is a common method for considering more variables without overfitting models. Chapter 0.4 will consider methods for incorporating regularization to incorporate these second-order terms.

0.3.4 Summary

Figure 13 illustrates how power-law forgetting is much more effective at modeling individual student performance in these data when it is modified to incorporate student-specific information. The overall results in Table 1 show a clear advantage to this approach, with the final power law forgetting model explaining an average 24.2% of the variance in activity-level errors. The second-order models Linear(f, f^2)and PL$_{\alpha(\text{f,f}^2),\beta(\text{f,f}^2)}$ show extremely high error and variability - this is due to overfitting on activities with few data points, leading to very high error on those activities.

Model Type	Student-Weighted Error	Activity-Weighted Error	Activity Error Std.
$\mathrm{PL}_{\alpha,\beta,\gamma}$	0.9373	0.8855	0.0762
$\mathrm{PL}_{\alpha,\beta}$	0.9378	0.8858	0.0763
$\mathrm{Linear(f)}$	0.7929	0.8001	0.0654
$\mathrm{PL}_{\alpha,\beta(\mathrm{f})}$	0.8356	0.8242	0.0601
$\mathrm{PL}_{\alpha(\mathrm{f}),\beta}$	0.7916	0.7994	0.0645
$\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$	0.7734	0.7581	0.0743
$\mathrm{Linear(f, f^2)}$	0.7763	0.8404	3.4218
$\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$	0.7875	0.9686	8.4507

Table 1: Summary of results for all models

0.4 Regularized Regression Models

In Chapter 0.3, the addition of second-order terms to the linear ($\text{Linear}(\text{f}, \text{f}^2)$) and power-law ($\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$) models demonstrated a well known shortcoming of least-squares regression: overfitting. The smallest activity has only 571 points, but the $\text{Linear}(\text{f}, \text{f}^2)$ model has 98 parameters and the $\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$ has 195, which is not much smaller than the number of data points in the smallest activity. The large number of parameters in these models made it likely that they overfit the activities with fewer data points, which led too poor test performance.

This problem was evidenced by a disparity between the student-weighted and activity-weighted errors. Activities with few data points would be overfit by the model, which caused the activity-weighted error to go up, while the student-weighted error was only slightly affected. Models with few parameters were able to predict the smaller activities well, but the more complex model had higher test errror. This problem can be addressed through the use of an L_1-norm regularized *Lasso* regression, described in Section 0.4.1.1. Another common method for addressing overfitting problems is to place a strong Bayesian prior on the coefficients of a linear regression. This technique is described in Section 0.4.1.2.

I evaluate both Lasso and Bayesian regularization for the models with many parameters that were overfit by least-squares regression ($\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$and $\text{Linear}(\text{f}, \text{f}^2)$). Both regularization methods are effective in dealing with the overfitting encountered in these models. Regularization was also applied to the models with few parameters, but did not improve their test accuracy.

Both regularization schemes are effective in dealing with the overfitting encountered. However, even with regularization, the second-order models fail to improve upon the performance of the $\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$ model. The results are presented in Section 0.4.2.

0.4.1 Regularization Methodology

0.4.1.1 L_1-Norm Regularized Linear Regression

When a model has a large number of regressors relative to the size of the data set, ordinary least-squares (henceforth, "OLS") regression can overfit the model, leading to a model with relatively poor accuracy in predicting new data points that are not part of the training set. OLS regression attempts to minimize the residual sum of squares for N coefficients $\beta_0..\beta_N$ given a regressand y and regressors $x_0...x_N$ [9]:

$$\hat{\beta}^{\text{OLS}} = \arg\min_{\beta} \sum (y - \beta_0 - \sum_{j=0}^{N} x_j \beta j)^2 \tag{20}$$

One reason for overfitting is the large number of free parameters in the model. To reduce a linear model's complexity, a standard approach in statistics is to incorporate a *regularization term* into Eq. 20 which penalizes unnecessary non-zero β coefficients by placing a limit on the sum of the absolute values of the β coefficients:

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \sum (y - \beta_0 - \sum_{j=0}^{N} x_j \beta j)^2$$
$$\text{Subject to} \sum_{j=1}^{N} |\beta_j| \leq t, \tag{21}$$

The t parameter sets a limit the L_1-norm of the β coefficients. This L_1 norm regularization has the effect of pushing β values to zero, effectively removing them from the regression. This type of regression is called the Lasso, and was first introduced by Tibshirani in [16].

The Lasso implementation used in this work is the `lasso` function in MATLAB, which is defined in terms the equivalent *Lagrangian* form of Eq. 21 [9]:

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum (y - \beta_0 - \sum_{j=0}^{N} x_j \beta_j)^2 + \lambda \sum_{j=0}^{N} |\beta_j| \right\} \tag{22}$$

In this form of the Lasso equation, the λ parameter represents the amount of regularization. Higher λ values will place a larger penalty on the sum of the coefficients, and correspond to a lower t in Eq. 21.

To determine the correct value for λ for each activity, I hold out 20% of the data points from each training set to create a validation set and find the λ value that minimizes validation error. To generate the test λ values, I used the `lasso` function's default geometric sequence of λ values, with the `NumLambdas` parameter set to 25, for speed reasons. I also set the `Alpha` parameter equal to 1.0 for pure Lasso (L_1-norm, rather than a mix of L_1- and L_2-norms) regression.

0.4.1.2 Bayesian Linear Regression

The standard multiple linear regression model can be written as

$$y = X\beta + \epsilon$$
$$\epsilon \sim N(0, \sigma^2), \tag{23}$$

where y represents the predicted variable, X is a matrix of features, β is a matrix of coefficients, and ϵ is Gaussian noise. The advantage of Bayesian inference is that with sensible priors, one can still draw reasonable conclusions with few observations. These Bayesian priors help to constrain the values of variables in the event that there are not enough data to constrain them. In the case of the overfit activities, the data do not provide a sufficient constraint on the β coefficients. I chose to treat the β coefficients as random variables with a prior distribution. The prior distribution I chose was multivariate Gaussian with a mean vector initialized to zero. The covariance matrix had its diagonal initialized to a very small constant ($\sqrt{0.026}$). Its non-diagonal entries were set to zero. I chose this constant variance by evaluating the $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$ model with many different hand-picked constants, and used the value with the best test error. In future research, I would treat the covariance of this prior distribution as a random variable and impose an even weaker hyperprior on it.

Predictions are made by marginalizing over the posterior on β, which takes into account both the prior distribution I defined and the likelihood of the training data. The model is trained using a Gibbs sampler, which is based on MATLAB code from [11] (Example 6.1). The Gibbs chain was run for 1100 total iterations, with the first 100 ignored as burn-in.

0.4.2 Results and Discussion

0.4.2.1 Addressing Overfitting

The overall results are shown in Figure 25. Each bar represents the mean activity-weighted test error for the model and fitting procedure it is labeled with. For the two second-order models I explored - Linear$(\mathrm{f}, \mathrm{f}^2)$ and $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$- the Lasso and Bayesian regularized fits outperformed Ordinary Least Squares (OLS). For the first-order models - Linear(f) and $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$- the regularized fits did not outperform OLS. Regularization correctly addressed the overfitting in models with many parameters, but models with fewer parameters showed no benefit from regularization.

Because each of the models tested in Figure 1 has a different number of free parameters, we can re-graph the data in Figure 25 by ordering the models by their complexity. Figure 26 shows the 12 models with the horizontal axis indicating number of free parameters in the model and the vertical axis indicating activity-weighted test error. Each line on the graph represents one fitting procedure.

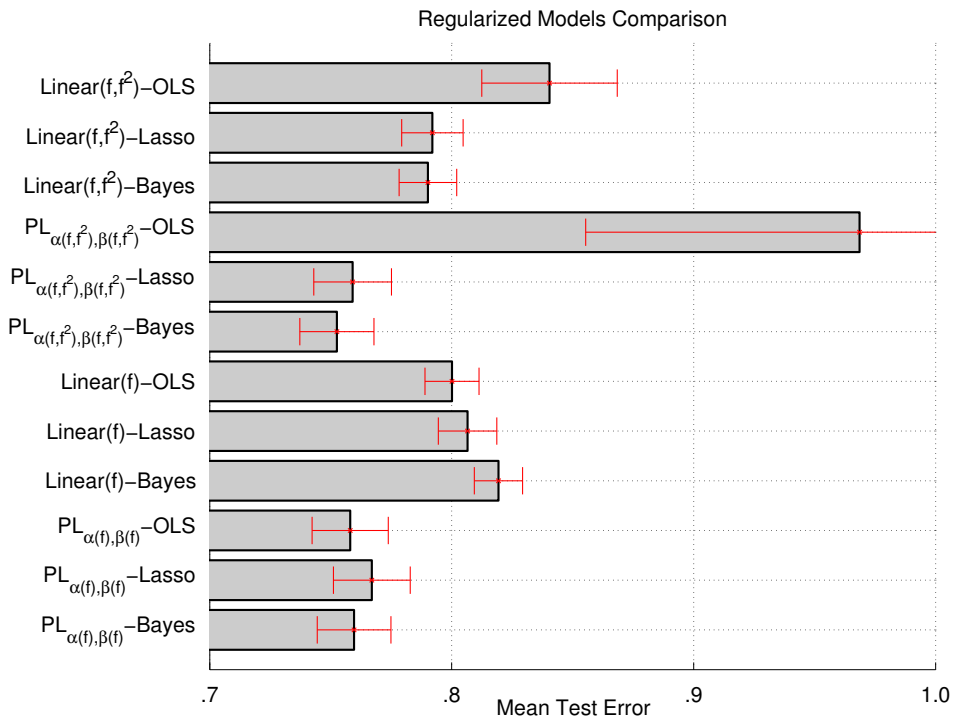

Figure 25: Mean activity test error for all models and fitting procedures considered in this Chapter. Note that, here "OLS" refers to "Ordinary Least Squares" regression. The error bars, in red, reflect within-activity variability, and have been corrected to remove between-activity variance as described in [14].

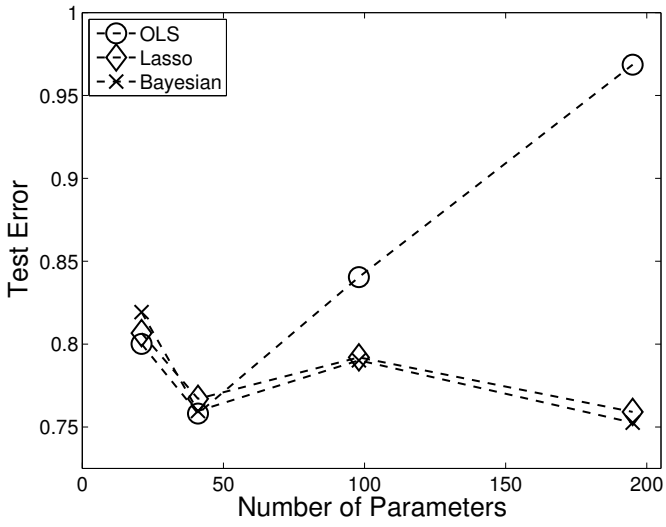

Figure 26: Test error for OLS, Lasso, and Bayesian fits as a function of the number of parameters. Each point on a line represents one model, from left to right: Linear(f), $\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$, Linear(f, f^2), $\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$.

The OLS-fit $\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$ model showed the most evidence of overfitting. I fit this model with both the Lasso and Bayesian regularization methods, both of which significantly improved test accuracy of the models, with smaller activities showing most of the gain. The Linear(f, f^2) model also showed overfitting. However, since it had less overfitting, the regularization methods did not give as much of a gain.

The Lasso fit for $\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$ had an activity-weighted error of 0.7591 vs 0.9686 for the OLS fit for $\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$ ($t(47) = -3.2559, p < 0.05$). This significant gain over the non-regularized OLS fit shows that it effectively addresses the overfitting problem. The detailed activity errors can be seen on the left in Figure 27. The differences between the OLS fit and the Lasso fit can be seen on the right. The trend in differences shows that activities with fewer numbers of data points have lower test error when fit with regularized regression.

The $\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$ model was also fit with the Bayesian sampling procedure, which also effectively addressed overfitting. The Bayesian fit had an activity-weighted error of 0.7526 vs 0.9686 for the OLS fitted $\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$ ($t(47) = -3.3784, p < 0.05$). The activity errors can be

seen in Figure 28 on the left, and the differences between the Bayesian and OLS fits can be seen on the right. Again, the trend shows that the smallest activities have the biggest reduction in error.

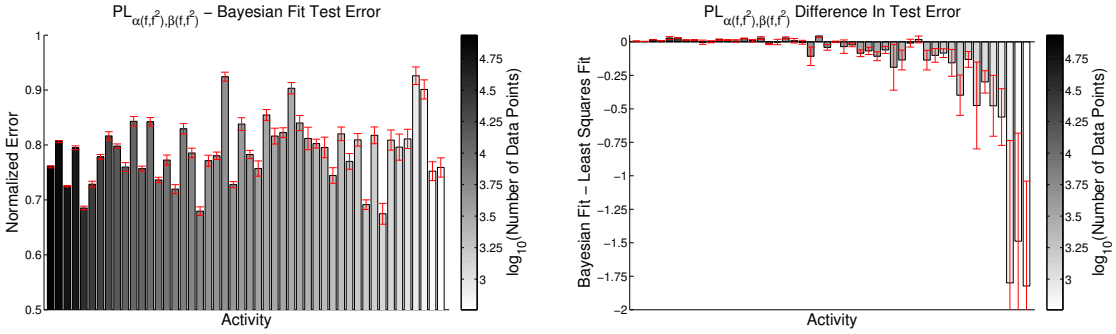

Figure 28: On the left, activity test errors for the Bayesian fit for the $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$ model. On the right, the differences in activity test errors between Bayesian- and OLS-fit models. Activities are sorted and colored by the number of data points, in decreasing order from left to right. The activities on the right have the fewest data points and show the greatest benefit from the Bayesian fitting method.

It is possible that even the simpler first-order models were also overfitting the data sets, but not as obviously as the more complex models. I tested both regularization methods on both Linear(f) and $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$, but none of the regularization methods showed a reduction in test error compared to OLS. This is consistent with the hypothesis that regularization is simply helping to address overfitting issues of the smaller activities for models with many parameters. It has either no effect or a detrimental effect when applied to the models with relatively few parameters.

0.4.2.2 Benefit of Second-Order Features

With the overfitting issues addressed, the second-order models have the potential to outperform the first-order models, since they are able to fit quadratic functions that the first-order models could not. However, the Lasso and Bayesian fits for the second-order $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$ model did not have a significant gain compared to the $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$ model, which had an activity-weighted error

of 0.7581. The t-tests comparing $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$to $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$failed to reach statistical significance for both Lasso ($t(47) = 0.5160, p = 0.6082$) and Bayesian ($t(47) = -2.6008, p = 0.0124$) fits for $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$. Regularization improved the test performance of the $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$ model, but it still did not outperform the simpler $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$ model.

0.4.3 Discussion

Both regularization procedures I used were effective in addressing the overfitting issues for OLS-fitted models with many parameters ($\mathrm{Linear}(\mathrm{f},\mathrm{f}^2)$and $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$). However, none of the regularized models significantly outperform $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$. This result implies that second-order terms do not help to explain any more of the variance in this data set than the individualized forgetting model coefficients do. Chapter 0.5 will go into detail on the coefficients of the $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$ model to try and understand what attributes are most predictive of student performance. The list of results for each model and regularization method are shown in Table 2.

Model	Fitting	Student-Weighted Err.	Activity-Weighted Err.	Activity Err. Std.
$\text{Linear}(\text{f}, \text{f}^2)$	OLS	0.7763	0.8404	3.4218
$\text{Linear}(\text{f}, \text{f}^2)$	Lasso	0.7781	0.7920	0.0685
$\text{Linear}(\text{f}, \text{f}^2)$	Bayesian	0.7731	0.7902	0.0625
$\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$	OLS	0.7875	0.9686	8.4507
$\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$	Lasso	0.7690	0.7591	0.0674
$\text{PL}_{\alpha(\text{f},\text{f}^2),\beta(\text{f},\text{f}^2)}$	Bayesian	0.7590	0.7526	0.0721
$\text{Linear}(\text{f})$	OLS	0.7929	0.8001	0.0654
$\text{Linear}(\text{f})$	Lasso	0.7969	0.8066	0.0608
$\text{Linear}(\text{f})$	Bayesian	0.7951	0.8193	0.0566
$\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$	OLS	0.7734	0.7581	0.0743
$\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$	Lasso	0.7803	0.7670	0.0667
$\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$	Bayesian	0.7751	0.7596	0.0738

Table 2: Summary of Lasso results for all models

0.5 Interpreting the Power Law Model

The best model considered for predicting student performance, $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$, makes use of a number of student-specific features, and was shown to reduce both student-weighted and activity-weighted error in Chapter 0.3. Next, I investigate the specifics of which features of the student and his study history are most useful in accurately predicting student performance. I will look at the coefficients on features of $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$, which was the best model explored, to understand which of its features helped it predict student performance. In order to directly compare the coefficients of features, I re-evaluated the models using features replaced with standard scores as described in Section 0.5.1. Since the features are referred to by their short names, the reader may find it helpful to refer back to the list of features in Chapter 0.1 while reading this Chapter.

0.5.1 Methodology for Model Interpretation

In order to understand which features of a regression model contribute the most to predictions, it is helpful to interpret the effect of features by inspecting their associated coefficients. A negative coefficient means that the feature associated with that coefficient has a negative relationship with the predicted variable, and vice versa. The strength of that relationship is represented by the magnitude of the coefficient. However, in order to compare the magnitudes of different coefficients to each other (e.g. in order to find the regressor with the most predictive power), it is important to first normalize the regressors so that they all have the same scale. For example, the coefficients for a score variable in the range $[0, 1]$ and for a time variable in the range $[0, +\infty]$ will have very different ranges - the coefficient on time is likely to be much smaller than the coefficient on the score. This does not necessarily imply that time is a less useful predictor than score.

To normalize the regressors to allow the direct comparison of coefficient values, the regressors for both the training and test sets are converted to standard scores by subtracting the mean of the training set μ_{train} and dividing by the standard deviation σ of the training set σ_{train}. For the

entire data set, each regressor x, a vector of data points, is replaced by its standard score $\hat{x}$:

$$\hat{x} = \frac{x - \mu_{train}}{\sigma_{train}} \tag{24}$$

This substitution is performed for all regressors and for the regressand. A linear regression model with regressors and regressand replaced with $\hat{x}$ is easily interpretable. The sign of the coefficients still reflects their relationship with the predicted variable, but their magnitudes are directly comparable.

0.5.2 Interpretation of Power Law Model Coefficients

Figure 29 shows a plot of the mean coefficients for the largest 25 coefficients corresponding to features of the power law model $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$. Each bar represents the mean coefficient for one feature across all 48 activities. Each bar is labeled with the shorthand name of that feature. A black bar indicates that the coefficient mean is negative, a white bar indicates that it is positive. The red error bars represent +/- 1 standard error across the 48 activities. Each activity's coefficients were calculated by taking the mean for all cross-validation splits of that activity. Recall that the linear form of $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$is:

$$\log s_2 = \sum_i^N \alpha_i' X_i + \sum_i^N \beta_i X_i \log t \tag{25}$$

Features that are part of the first summation (the $\alpha(\mathrm{f})$ function) are marked with their shorthand names. Features that are part of the second summation (the $\beta(\mathrm{f})$ function) are marked **log(TimeDelta)*FeatureName**. The 25 largest mean coefficients are plotted in Figure 29. The largest coefficients contribute the most to predicting $\log s_2$.

Unsurprisingly, the intercept term of the $\beta(\mathrm{f})$ function, **log(TimeDelta)**, is the feature that contributes most to $\log s_2$ in the $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$model. This term represents the basic forgetting curve of the activity, before taking into account the other student-specific factors on forgetting. Figure 30 shows individual log (TimeDelta) coefficient values for each activity. In Figure 29, there is a large variance associated with this feature. There is a large spread of default forgetting in these activities. Figure 30 shows the spread of this parameter across all 48 activities, ordered by index in

the curriculum. It is clear that the rates of forgetting have strong per-activity variation. Notably, the linear version of **TimeDelta** is much weaker.

Another strong predictor of student performance, according to the $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$ model, is **LessonActivityTypeCount**, the count of types of activities in the lesson that were completed before attempting $\log s_2$. Its negative relationship with $\log s_2$ indicates that students who do more different types of activities in the intervening time before attempting $\log s_2$ again tend to have lower scores when re-tested. This seems counter-intuitive: students who practiced the material in the lesson more should, in theory, perform better on the review activity that represents that lesson. A possible explanation for this strange relationship is that a student who opts for more optional activities consumes more content in general, and thus shows interference from learning content in other lessons.

The next strong predictor is also part of the $\beta(\mathrm{f})$ function: **log(TimeDelta)*Score1**. It is positively correlated to $\log s_2$, which indicates that learners who perform better on their first attempt (a higher s_1) tend to forget material at a slower rate than learners who performed better initially. It has a a variance similar to that of **log(TimeDelta)**. The coefficients on this feature are plotted in Figure 30 on the right.

The next strongest coefficient corresponds to **Score1**, the score on the first attempt of the review activity. It is notable that this feature, despite being the one most obviously related to $\log s_2$, does not have the strongest relationship to $\log s_2$. It has an unsurprising positive relationship with $\log s_2$. Similarly, the score on the "core lesson" (**CoreScore**) has a positive, but weaker, relationship with $\log s_2$.

The binary variables indicating the that the learner opted to attempt a particular special activity in that lesson (e.g. **DidListeningAndReading**, **DidWriting**, etc.) were also very strong in general. They are all positively correlated with $\log s_2$. This is interesting since a related variable, **LessonActivityTypeCount**, is highly negatively correlated. A possible explanation is that opting for some additional content is good and reinforces the material in that lesson, but being a very heavy user of the product causes more interference and will lead to more rapid forgetting.

Several other features were strongly related to $\log s_2$. The negative coefficient on **log(TimeDelta)* LessonActivityTypeCount** indicates that LessonActivityTypeCount also has the effect increasing the rate of forgetting, in addition to the absolute amount forgottten.

The features related to partial attempts on the review activity, **PartialTime**, and **PartialCount**, seem to contradict one another. **PartialTime** is the amount of time spent on incomplete/partial attempts of the review activity, and is weakly negatively related to $\log s_2$. **PartialCount** is the number of such partial attempts, but is positively related to $\log s_2$. A further investigation could help tease apart why these variables, which should be highly correlated, seem to have opposite effects on s_2.

Another predictor, weakly negatively related to $\log s_2$, is **SecondIsAR**, a binary variable indicating whether the second attempt is scheduled automatically by the system. This indicator variable divides the population into two groups: students who reviewed the activity immediately and students who waited to review. As noted in Chapter 0.1, the system begins to recommend to review an activity two weeks after the initial attempt. Therefore, when the second attempt was scheduled by the system, it means that at least two weeks have passed. The coefficient on **SecondIsAR** is negative because more forgetting happens after two weeks than happens before two weeks. Recall the bimodal histogram of time intervals in Figure 7 from Chapter 0.1. This feature serves to separate students in the first bump from students in the second. Of course, the model has a more direct representation of time in the **DeltaTime** feature. Why is the coefficient on **SecondIsAR** stronger than the coefficient on **TimeDelta**? One possible explanation is that the linear representation of time is not appropriate to establish a linear relationship with $\log s_2$, and this binary variable is a means of separating two groups with a clear separation in logarithmic time.

Other features related to additional use of the product within a lesson were negatively related to $\log s_2$. **LessonActivityScoreStd** represents one standard deviation of the activity scores within the lesson, and is weakly negatively related to $\log s_2$. **LessonActivityTime** represents the amount of time spent on all other activities within the lesson that are not the review activity. It is negatively,

albeit weakly, related to the predicted variable. A detailed study investigating the individual activity scores on the review activity might help shed light on why these features seem negatively related to $\log s_2$, but is out of the scope of this thesis.

0.5.3 Summary

The constituent features of the forgetting rate function β(f) (the terms multiplied by **log(TimeDelta)**) play a significant role in predicting student performance. The intercept of this β(f) function (**log(TimeDelta)**) represents the default rate of forgetting for an activity - this was the strongest mean coefficient. The log(TimeDelta) interaction terms have the function of modifying the rate of forgetting (the shape of the forgetting function) for a particular student based on her study history. Several of these, such as **log(TimeDelta)* Score1**, have a significant effect of customizing the forgetting curve for a particular student.

Many of the terms comprising the α(f) function also have an effect: notably, their initial score on the review activity and a number of indicator variables describing the details of the student's study history in that lesson. These terms affect the scale of the forgetting function, but not its shape.

0.6 Conclusions

Predictions of student performance for review activities in the Rosetta Stone® course are most accurate when they take into account both power-law forgetting and the student's study history.

The two-parameter power law has the same ability to describe forgetting in these data as the three-parameter power law. The two-parameter model can be fitted using least-squares linear regression, which allows both parameters of this model to be replaced with linear functions that combine features of an individual student's study history. Compared to predictions made from the two-parameter model, this individualized model makes predictions that are 14% better activity-weighted, and 17% better student-weighted. Compared to simple linear regression which does not take power-law forgetting into account, the predictions are 5% better activity-weighted and 2% better student-weighted. The addition of second-order terms to this individualized power-law model had no significant effect on activity-weighted prediction accuracy.

Compared to a more traditional machine learning approach, this model is rooted in the psychological theory of forgetting. This is valuable because the parameters of this model have clear interpretations that can explain the factors influencing forgetting in the Rosetta Stone® course. This knowledge can be used to improve the product to increase retention and to make accurate predictions of performance.

There are several implications of this work for the Rosetta Stone® product. The existing Adaptive Recall® review function already makes suggestions for students to review activities after a time interval. This feature could be enhanced to make review suggestions for time intervals that are based on individualized estimates of the student's forgetting over time. For example, the system could recommend that a student review an activity right before she is estimated to drop below the passing score threshold.

As students continue through the product, they tend to accumulate a queue of review activities to look at. If a student has a large queue, he is not likely to go through the entire set of

review suggestions in that session. However, the individualized forgetting model could be used to prioritize the queue and recommend only the one activity for which immediate study would have the greatest impact on retention. With such a feature, students pressed for time could maximize the impact of a short study session.

Students could be given a dashboard view of their own estimated retention for each review activity. Such a feature might help motivate students who are unaware of their own forgetting to review the activities more often. It could also help teachers track the memory of a whole classroom of students.

0.6.1 Future Work

The progression of models as introduced in Chapter 0.3, Fig. 10 can be interpreted to represent a set of points in a *space* of models. If we view the power-law models as linear regressions in log space, as suggested by our training procedure in Eq. 17, then the only difference between the power-law model $\text{PL}_{\alpha(\text{f}),\beta}$and the simple linear model Linear(f)is the prediction space: $log(s_2)$ or s_2, respectively. The prediction type is the first dimension in the model space.

Suppose we define α simply as a term linearly related to whatever prediction we make. It can be composed of a single term (α) as in the two-parameter model, or it can be composed of a linear combination of features ($\alpha(\text{f})$), as in the linear regression model and in $PL_{\alpha(\text{f}),\beta}$.

Similarly, we can define β as some term multiplied by $\log(t)$. Again, β has two variants: the single-coefficient version (β) and the linear combination version ($\beta(\text{f})$). As such, the inclusion of α or $\alpha(\text{f})$ and β or $\beta(\text{f})$ can be viewed as dimensions in this model space. Note that α and $\alpha(\text{f})$ are mutually exclusive, since $\alpha(\text{f})$ is a proper superset of α, and similarly for β and $\beta(\text{f})$. Under this terminology, the simple linear model is named $\text{Linear}_{\alpha(\text{f})}$.

Prediction	α	β	α(f)	β(f)	Name	Chapter Name
$\log(s_2)$	✓				$\text{logPredict}_{\alpha}$	
$\log(s_2)$		✓			$\text{logPredict}_{\beta}$	
$\log(s_2)$	✓	✓			$\text{logPredict}_{\alpha,\beta}$	$\text{PL}_{\alpha,\beta}$
$\log(s_2)$			✓		$\text{logPredict}_{\alpha(\text{f})}$	
$\log(s_2)$		✓	✓		$\text{logPredict}_{\alpha(\text{f}),\beta}$	$\text{PL}_{\alpha(\text{f}),\beta}$
$\log(s_2)$				✓	$\text{logPredict}_{\beta(\text{f})}$	
$\log(s_2)$	✓			✓	$\text{logPredict}_{\alpha,\beta(\text{f})}$	$\text{PL}_{\alpha,\beta(\text{f})}$
$\log(s_2)$			✓	✓	$\text{logPredict}_{\alpha(\text{f}),\beta(\text{f})}$	$\text{PL}_{\alpha(\text{f}),\beta(\text{f})}$
s_2	✓				$\text{linearPredict}_{\alpha}$	
s_2		✓			$\text{linearPredict}_{\beta}$	
s_2	✓	✓			$\text{linearPredict}_{\alpha,\beta}$	
s_2			✓		$\text{linearPredict}_{\alpha(\text{f})}$	Linear(f)
s_2		✓	✓		$\text{linearPredict}_{\alpha(\text{f}),\beta}$	
s_2				✓	$\text{linearPredict}_{\beta(\text{f})}$	
s_2	✓			✓	$\text{linearPredict}_{\alpha,\beta(\text{f})}$	
s_2			✓	✓	$\text{linearPredict}_{\alpha(\text{f}),\beta(\text{f})}$	

Table 3: A space of models suggested by permuting the combinations of models explored in this work.

Viewing the permuted model space in Table 3 makes it obvious that there are many more combinations of models to explore. In future work, it maybe informative to investigate how well some of the other permutations of these functions may describe these data, keeping in mind that some of these permutations may not have a clear interpretation in terms of psychological theory.

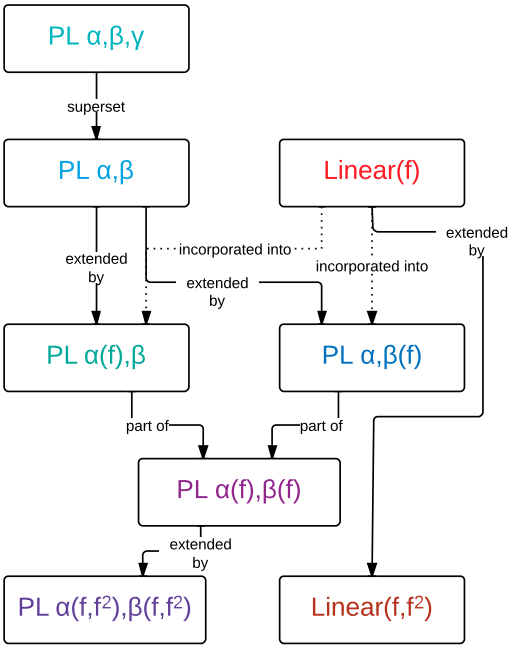

Figure 10: Hierarchy of models explored. First, we will discuss the three- and two-parameter power law models $\mathrm{PL}_{\alpha,\beta,\gamma}$and $\mathrm{PL}_{\alpha,\beta}$introduced in Chapter 0.1. These models are compared with a linear regression $\mathrm{Linear(f)}$, which incorporates individual-specific features to make predictions of performance. The linear combination of features used in $\mathrm{Linear(f)}$is incorporated into the two-parameter power-law model $\mathrm{PL}_{\alpha,\beta}$. The two-parameter model is extended by using the linear combination of features to estimate its β term ($\mathrm{PL}_{\alpha,\beta(\mathrm{f})}$), its α term ($\mathrm{PL}_{\alpha(\mathrm{f}),\beta}$), or both ($\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$). Finally, second-order terms are added to $\mathrm{Linear(f)}$ to create $\mathrm{Linear}(\mathrm{f},\mathrm{f}^2)$, and to $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$ to create $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$.

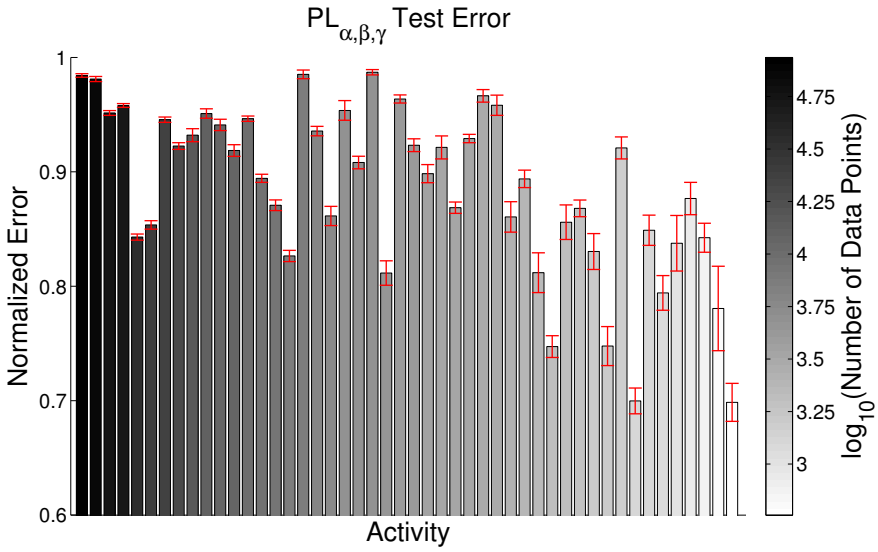

Figure 11: The mean normalized three-parameter forgetting error values for the test sets of each of the 48 unique activities in the data set. Each activity is represented by a bar whose height indicates the normalized error. The red bars indicate +/- 1 standard error of the mean, computed across cross validation splits of the data. The coloring of a bar indicates the size of the data set for a given activity, and the activities are ordered from most to least data. The most popular activity had 86,296 data points, the least popular only 571. Note that there seems to be a general trend towards lower error for activities with fewer data points ($R^2 = 0.17$). This trend will be discussed in further detail later in the chapter.

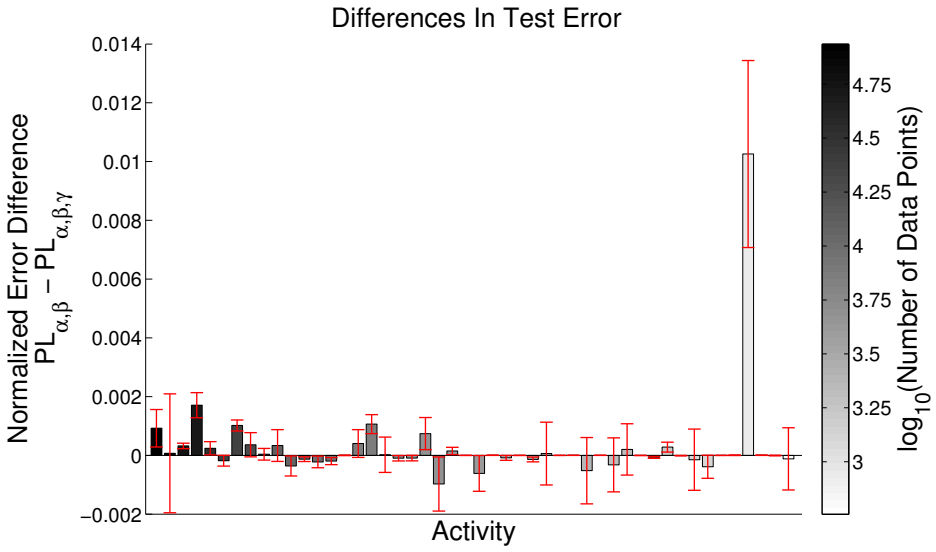

Figure 12: The differences in normalized error between $\mathrm{PL}_{\alpha,\beta}$and $\mathrm{PL}_{\alpha,\beta,\gamma}$. The vertical axis represents the difference in error between the two models. Each bar represents the difference in error for one activity. A positive difference for an activity indicates that the error on $\mathrm{PL}_{\alpha,\beta}$ is higher, a negative difference indicates that the $\mathrm{PL}_{\alpha,\beta,\gamma}$ error is higher. The red error bars represent +/- one standard error of the differences between the cross validation splits.

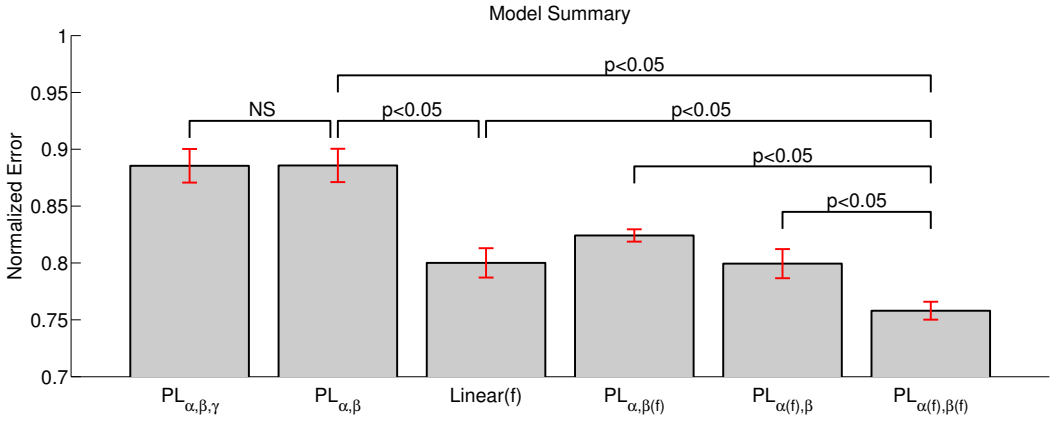

Figure 13: Summary of error for all models considered. Significant differences between models are marked "$p < 0.05$", and non-significant differences are marked "NS". The error bars, in red, reflect within-activity variability, and have been corrected to remove between-activity variance as described in [14]. Note that $\mathrm{Linear}(\mathrm{f}, \mathrm{f}^2)$ and $\mathrm{PL}_{\alpha(\mathrm{f},\mathrm{f}^2),\beta(\mathrm{f},\mathrm{f}^2)}$ have been omitted from this graph due to their high error, discussed later in the chapter.

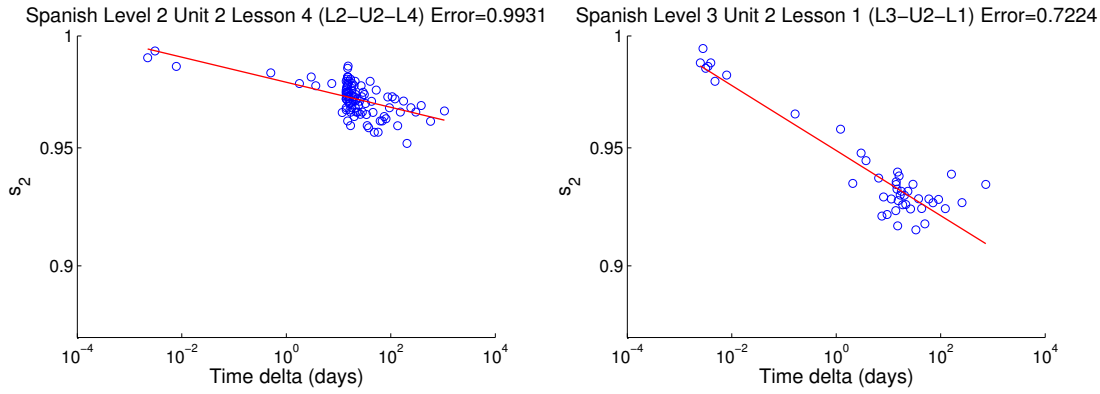

Figure 18: Comparison of one of the worst-fitting activities, left, to one of the best-fitting activities, right. Data are binned with each bin containing 50 data points. The normalized error on the raw data points is reported for each activity.

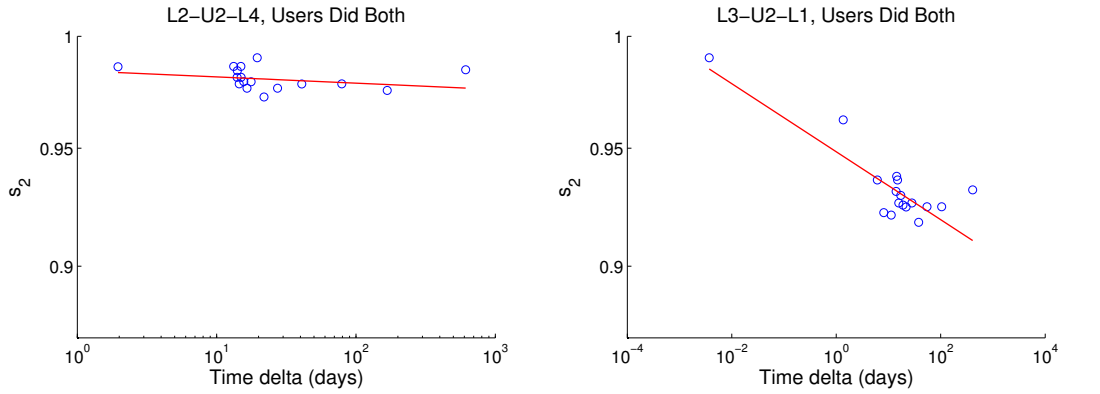

Figure 19: Power-Law forgetting plots for L2-U2-L4 and L3-U2-L1, including only learners who did both activities.

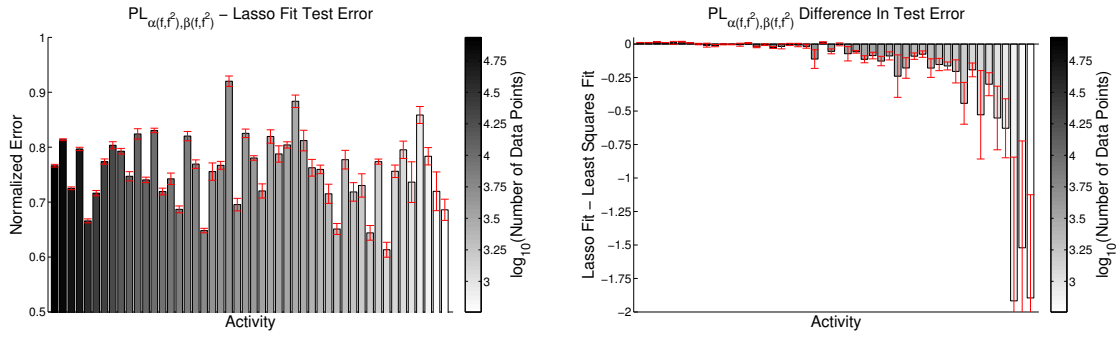

Figure 27: On the left, activity test errors for the Lasso-fit $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$ model. On the right, the differences in activity test errors between Lasso and OLS fit models. Activities are sorted and colored by the number of data points, in decreasing order from left to right. The activities on the right have the fewest data points and show the greatest benefit from the Lasso fitting method.

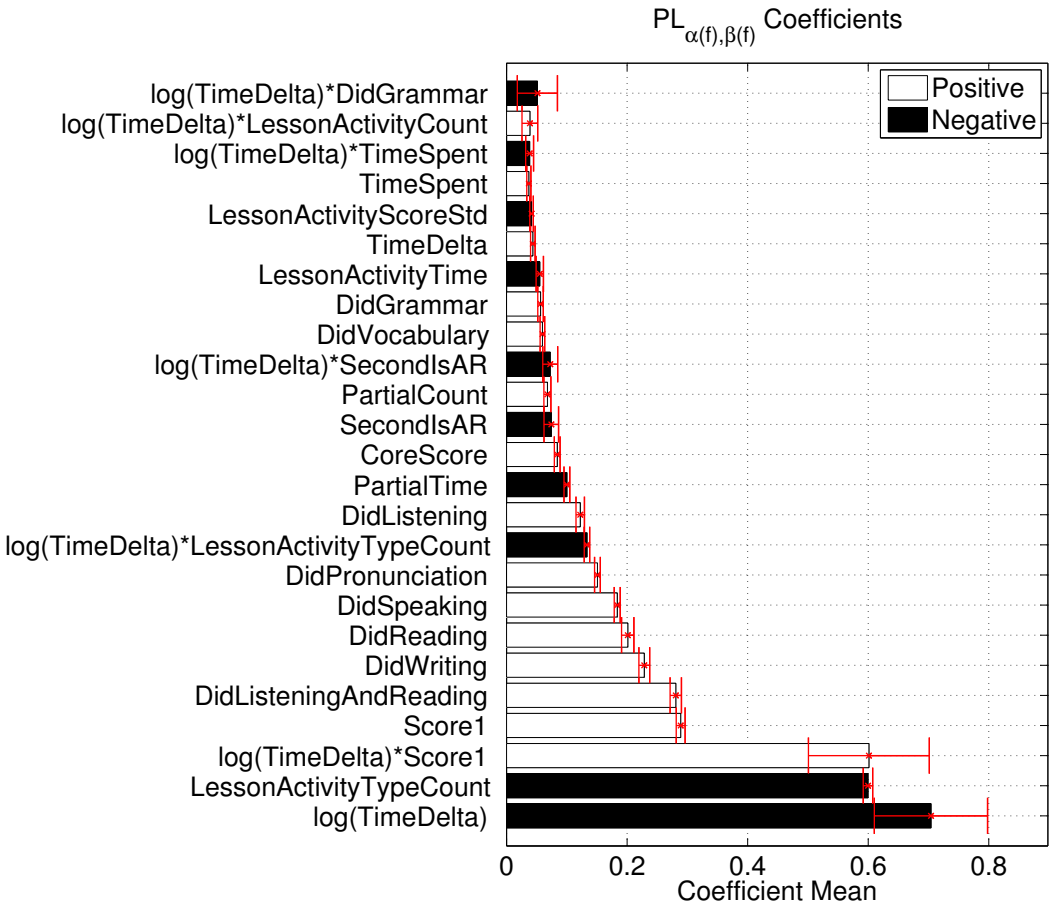

Figure 29: Mean coefficient values for the 25 coefficients of $\mathrm{PL}_{\alpha(\mathrm{f}),\beta(\mathrm{f})}$ with the largest magnitudes, sorted top-down in increasing order of the absolute value of the coefficient, minus one standard error. These coefficient values are estimated with standard-score features, described in Section 0.5.1. A larger coefficient denotes that that variable has a stronger relationship with the predicted variable, $\log s_2$. The error bars, in red, represent +/- 1 standard error across activities.

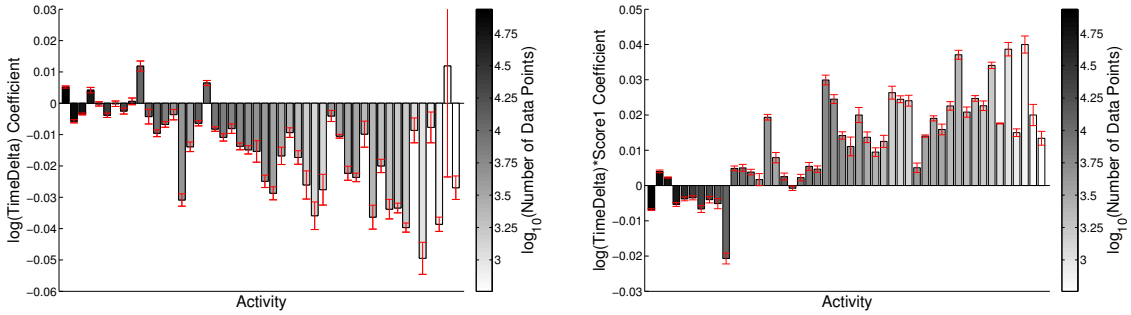

Figure 30: On the left, the **log(TimeDelta)** coefficient for all activities, in order of the appearance in the curriculum. On the right, the **log(TimeDelta)*Score1** coefficient for all activities, in curriculum order. Error bars, in red, represent +/- one standard error between cross-validation splits. Note that the coefficients in his graph are not based on the standard-score models built for comparing coefficients. Their ranges reflect their actual values when predicting $\log_{s_2}$

Bibliography

[1] Harry P Bahrick. The cognitive map of a city: Fifty years of learning and memory. The Psychology of Learning and Motivation: Advances in Research and Theory, 17:125–163, 1983.

[2] Harry P Bahrick, Lorraine E Bahrick, Audrey S Bahrick, and Phyllis E Bahrick. Maintenance of foreign language vocabulary and the spacing effect. Psychological Science, 4(5):316–321, 1993.

[3] Harry P Bahrick and Lynda K Hall. Lifetime maintenance of high school mathematics content. Journal of Experimental Psychology: General, 120(1):20, 1991.

[4] R. Bjork. Memory and metamemory considerations in the training of human beings. In Metacognition: Knowing about knowing, pages 195–205. MIT Press, 1994.

[5] Kristine C Bloom and Thomas J Shuell. Effects of massed and distributed practice on the learning and retention of second-language vocabulary. Journal of Educational Research, 74(4):245–48, 1981.

[6] Eugene Custers. Long-term retention of basic science knowledge: a review study. Advances in Health Science Education: Theory & Practice, 15(1):109–128, 2010.

[7] Eugene Custers and O. ten Cate. Very long-term retention of basic science knowledge in doctors after graduation. Medical Education, 45(4):422–430, 2011.

[8] Hermann Ebbinghaus. Über das Gedächtnis. Untersuchungen zur experimentellen Psychologie. Leipzig: Duncker & Humblot, 1885.

[9] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. The elements of statistical learning, volume 2. Springer, 2009.

[10] G. Keim. Adaptive recall, March 5 2009. US Patent App. 12/052,435.

[11] James P LeSage. Applied econometrics using matlab. Manuscript, Dept. of Economics, University of Toronto, pages 154–159.

[12] R. Lindsey, J. Shroyer, H. Pashler, and M. Mozer. Improving students' long-term knowledge retention through personalized review. Psychological Science, 2014.

[13] Scott E Lively, David B Pisoni, Reiko A Yamada, Yoh'ichi Tohkura, and Tsuneo Yamada. Training japanese listeners to identify english/r/and/l/. iii. long-term retention of new phonetic categories. The Journal of the Acoustical Society of America, 96(4):2076–2087, 1994.

[14] Michael EJ Masson and Geoffrey R Loftus. Using confidence intervals for graphically based data interpretation. Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale, 57(3):203, 2003.

[15] David C Rubin and Amy E Wenzel. One hundred years of forgetting: A quantitative description of retention. Psychological Review, 103(4):734–760, 1996.

[16] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pages 267–288, 1996.

[17] J.T. Wixted and S.K. Carpenter. The wickelgren power law and the ebbinghaus savings function. Psychological Science, 18:133–134, 2007.

[18] Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rose. "turn on, tune in, drop out": Anticipating student dropouts in massive open online courses.